

Tivoli Application Dependency Discovery  
Manager  
Version 7.3

*User's Guide*



**Note**

Before using this information and the product it supports, read the information in [“Notices” on page 245.](#)

**Edition notice**

This edition applies to version 7, release 3 of IBM® Tivoli® Application Dependency Discovery Manager (product number 5724-N55) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2006, 2020.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Figures.....</b>	<b>v</b>
<b>Tables.....</b>	<b>vii</b>
<b>About this information.....</b>	<b>ix</b>
Conventions used in this information center.....	ix
Terms and definitions.....	ix
<b>Chapter 1. Using.....</b>	<b>1</b>
Discovery Management Console.....	1
Starting the Discovery Management Console.....	1
Discovery scopes.....	2
Access lists.....	5
Running discoveries.....	11
Managing discoveries.....	13
Reconciling configuration items.....	74
Data Management Portal.....	78
Discovery tasks.....	78
Topology tasks.....	85
Analytics tasks.....	89
Administration tasks.....	97
Domain management tasks.....	100
Data Access Portal.....	106
Logging In.....	106
Signing out.....	107
Dashboard pane.....	107
Search.....	108
Performing suggestive search.....	108
Performing normal search.....	108
Performing advance search.....	108
Viewing component details.....	109
Comparing components.....	109
Viewing Relationship.....	110
Details pane.....	111
View comparison pane.....	111
Component Comparison: result pane.....	112
Relationship pane.....	112
User interface reference.....	112
Discovery Management Console windows and controls.....	113
Data Management Portal windows and controls.....	131
Task scenarios.....	168
Setting up a discovery.....	168
Extending custom servers.....	173
Business Applications.....	175
Getting started with business applications.....	176
Business application structure.....	177
Creating business applications with grouping patterns.....	179
Creating business applications with application descriptors.....	183
Creating business applications with Java API.....	187

Displaying business applications.....	198
Processing of grouping patterns.....	200
bizappscli tool.....	216
Configuring the collation .properties file entries.....	231
Logging.....	233
Migration from 7.2.2 and automatic conversion of old business applications.....	233
Integrating business applications with other Tivoli products.....	237
Example scenarios.....	237
<b>Notices.....</b>	<b>245</b>
Trademarks.....	246

---

# Figures

1. Attribute Prioritization window..... 76

2. Topology with the HigherUp option selected..... 213

3. Topology with the HigherUp and HigherDown options selected..... 213

4. Topology with the LowerDown option selected..... 214

5. Topology with the LowerDown and LowerUp options selected..... 214

6. Topology with only HigherUp option selected.....215

7. Topology with HigherUp and HigherDown options selected..... 215

8. Topology with only LowerDown option selected.....216

9. Topology with LowerDown and LowerUp options selected..... 216



---

# Tables

1. Discovery scope information.....	2
2. Required component types, fields, and lists for access list entry .....	6
3. Predefined custom server templates.....	20
4. Directive file format.....	31
5. Directive file environment variables.....	32
6. Target map objects for the custom server.....	34
7. Target map objects for the computer system .....	35
8. Difference between Custom Template Sensor (CTS), Custom Server Template (CST), and Custom Server Extension (CSX).....	40
9. Discovery history information.....	50
10. BiDi attributes.....	62
11. Base Application Descriptor elements and attributes.....	67
12. Component application descriptor elements and attributes.....	68
13. Default Application Descriptor Locations.....	70
14. Application description.....	71
15. Specialized topologies.....	86
16. Specialized topologies.....	106
17. ....	108
18. Items in the Discovery tab.....	113
19. ....	116
20. Required component types, fields, and lists for access list entry .....	116
21. Topology toolbar tool icons .....	142
22. Business applications pop-up menu items.....	143

23. Routes tab details.....	144
24. Grouping pattern API methods.....	189
25. Pattern schedule management.....	190
26. Pattern execution management methods.....	191
27. Tiers configuration elements and attributes.....	208



## About this information

---

The purpose of this PDF document version is to provide the related topics from the information center in a printable format.

## Conventions used in this information center

---

In the IBM Tivoli Application Dependency Discovery Manager (TADDM) documentation certain conventions are used. They are used to refer to the operating system-dependent variables and paths, the `COLLATION_HOME` directory, and the location of the `collation.properties` file, which is referenced throughout the TADDM documentation, including in the messages.

### Operating system-dependent variables and paths

In this information center, the UNIX conventions are used for specifying environment variables and for directory notation.

When using the Windows command line, replace `$variable` with `%variable%` for environment variables, and replace each forward slash (/) with a backslash (\) in directory paths.

If you are using the bash shell on a Windows system, you can use the UNIX conventions.

### COLLATION\_HOME directory

TADDM root directory is also referred to as the `COLLATION_HOME` directory.

On operating systems such as AIX® or Linux®, the default location for installing TADDM is the `/opt/IBM/taddm` directory. Therefore, in this case, the `$COLLATION_HOME` directory is `/opt/IBM/taddm/dist`.

On Windows operating systems, the default location for installing TADDM is the `c:\IBM\taddm` directory. Therefore, in this case, the `%COLLATION_HOME%` directory is `c:\IBM\taddm\dist`.

### Location of collation.properties file

The `collation.properties` file contains TADDM server properties and includes comments about each of the properties. It is located in the `$COLLATION_HOME/etc` directory.

## Terms and definitions

---

Refer to the following list of terms and definitions to learn about important concepts in the IBM Tivoli Application Dependency Discovery Manager (TADDM).

#### access collection

A collection that is used to control the access to configuration items and permissions to modify configuration items. You can create access collections only when data-level security is enabled.

#### asynchronous discovery

In TADDM, the running of a discovery script on a target system to discover systems that cannot be accessed directly by the TADDM server. Because this discovery is performed manually, and separately from a typical credentialed discovery, it is called "asynchronous".

#### business application

A collection of components that provides a business functionality that you can use internally, externally, or with other business applications.

#### CI

See *configuration item*.

#### collection

In TADDM, a group of configuration items.

**configuration item (CI)**

A component of IT infrastructure that is under the control of configuration management and is therefore subject to formal change control. Each CI in the TADDM database has a persistent object and change history associated with it. Examples of a CI are an operating system, an L2 interface, and a database buffer pool size.

**credentialled discovery**

TADDM sensor scanning that discovers detailed information about the following items:

- Each operating system in the runtime environment. This scanning is also known as Level 2 discovery, and it requires operating system credentials.
- The application infrastructure, deployed software components, physical servers, network devices, virtual systems, and host data that are used in the runtime environment. This scanning is also known as Level 3 discovery, and it requires both operating system credentials and application credentials.

**credential-less discovery**

TADDM sensor scanning that discovers basic information about the active computer systems in the runtime environment. This scanning is also known as Level 1 discovery, and it requires no credentials.

**Data Management Portal**

The TADDM web-based user interface for viewing and manipulating the data in a TADDM database. This user interface is applicable to a domain server deployment, to a synchronization server deployment, and to each storage server in a streaming server deployment. The user interface is very similar in all deployments, although in a synchronization server deployment, it has a few additional functions for adding and synchronizing domains.

**discover worker thread**

In TADDM, a thread that runs sensors.

**Discovery Management Console**

The TADDM client user interface for managing discoveries. This console is also known as the Product Console. It is applicable to a domain server deployment and to discovery servers in a streaming server deployment. The function of the console is the same in both of these deployments.

**discovery server**

A TADDM server that runs sensors in a streaming server deployment but does not have its own database.

**domain**

In TADDM, a logical subset of the infrastructure of a company or other organization. Domains can delineate organizational, functional, or geographical boundaries.

**domain server**

A TADDM server that runs sensors in a domain server deployment and has its own database.

**domain server deployment**

A TADDM deployment with one domain server. A domain server deployment can be part of a synchronization server deployment.

In a domain server deployment, the following TADDM server property must be set to the following value:

```
com.collation.cmdbmode=domain
```

**launch in context**

The concept of moving seamlessly from one Tivoli product UI to another Tivoli product UI (either in a different console or in the same console or portal interface) with single sign-on and with the target UI in position at the proper point for users to continue with their task.

**Level 1 discovery**

TADDM sensor scanning that discovers basic information about the active computer systems in the runtime environment. This scanning is also known as credential-less discovery because it requires no credentials. It uses the Stack Scan sensor and the IBM® Tivoli® Monitoring Scope sensor. Level 1 discovery is very shallow. It collects only the host name, operating system name, IP address, fully

qualified domain name, and Media Access Control (MAC) address of each discovered interface. Also, the MAC address discovery is limited to Linux on System z® and Windows systems. Level 1 discovery does not discover subnets. For any discovered IP interfaces that do not belong to an existing subnet that is discovered during Level 2 or Level 3 discovery, new subnets are created based on the value of the `com.collation.IpNetworkAssignmentAgent.defaultNetmask` property in the `collation.properties` file.

### Level 2 discovery

TADDM sensor scanning that discovers detailed information about each operating system in the runtime environment. This scanning is also known as credentialed discovery, and it requires operating system credentials. Level 2 discovery collects application names and the operating system names and port numbers that are associated with each running application. If an application has established a TCP/IP connection to another application, this information is collected as a dependency.

### Level 3 discovery

TADDM sensor scanning that discovers detailed information about the application infrastructure, deployed software components, physical servers, network devices, virtual systems, and host data that are used in the runtime environment. This scanning is also known as credentialed discovery, and it requires both operating system credentials and application credentials.

### multitenancy

In TADDM, the use by a service provider or IT vendor of one TADDM installation to discover multiple customer environments. Also, the service provider or IT vendor can see the data from all customer environments, but within each customer environment, only the data that is specific to the respective customer can be displayed in the user interface or viewed in reports within that customer environment.

### Product Console

See *Discovery Management Console*.

### script-based discovery

In TADDM, the use, in a credentialed discovery, of the same sensor scripts that sensors provide in support of asynchronous discovery.

### SE

See *server equivalent*.

### server equivalent (SE)

A representative unit of IT infrastructure, defined as a computer system (with standard configurations, operating systems, network interfaces, and storage interfaces) with installed server software (such as a database, a web server, or an application server). The concept of a server equivalent also includes the network, storage, and other subsystems that provide services to the optimal functioning of the server. A server equivalent depends on the operating system:

Operating system	Approximate number of CIs
Windows	500
AIX	1000
Linux	1000
HP-UX	500
Network devices	1000

### storage server

A TADDM server that processes discovery data that is received from the discovery servers and stores it in the TADDM database. The primary storage server both coordinates the discovery servers and all other storage servers and serves as a storage server. All storage servers that are not the primary are called secondary storage servers.

### **streaming server deployment**

A TADDM deployment with a primary storage server and at least one discovery server. This type of deployment can also include one or more optional secondary storage servers. The primary storage server and secondary storage servers share a database. The discovery servers have no database.

In this type of deployment, discovery data flows in parallel from multiple discovery servers to the TADDM database.

In a streaming server deployment, the following TADDM server property must be set to one of the following values:

- `com.collation.taddm.mode=DiscoveryServer`
- `com.collation.taddm.mode=StorageServer`

For all servers except for the primary storage server, the following properties (for the host name and port number of the primary storage server) must also be set:

- `com.collation.PrimaryStorageServer.host`
- `com.collation.PrimaryStorageServer.port`

If the `com.collation.taddm.mode` property is set, the `com.collation.cmdbmode` property must not be set or must be commented out.

### **synchronization server**

A TADDM server that synchronizes discovery data from all domain servers in the enterprise and has its own database. This server does not discover data directly.

### **synchronization server deployment**

A TADDM deployment with a synchronization server and two or more domain server deployments, each of which has its own local database.

In this type of deployment, the synchronization server copies discovery data from multiple domain servers one domain at a time in a batched synchronization process.

In a synchronization server deployment, the following TADDM server property must be set to the following value:

```
com.collation.cmdbmode=enterprise
```

This type of deployment is obsolete. Therefore, in a new TADDM deployment where more than one server is needed, use the streaming server deployment. A synchronization server can be converted to become a primary storage server for a streaming server deployment.

### **TADDM database**

In TADDM, the database where configuration data, dependencies, and change history are stored.

Each TADDM server, except for discovery servers and secondary storage servers, has its own database. Discovery servers have no database. Storage servers share the database of the primary storage server.

### **TADDM server**

A generic term that can represent any of the following terms:

- domain server in a domain server deployment
- synchronization server in a synchronization server deployment
- discovery server in a streaming server deployment
- storage server (including the primary storage server) in a streaming server deployment

### **target system**

In the TADDM discovery process, the system to be discovered.

### **utilization discovery**

TADDM sensor scanning that discovers utilization information for the host system. A utilization discovery requires operating system credentials.

---

# Chapter 1. Using

## Discovery Management Console

---

The Discovery Management Console is the IBM Tivoli Application Dependency Discovery Manager (TADDM) client user interface for managing discoveries. This console is also known as the Product Console. It is applicable to a domain server deployment and to discovery servers in a streaming server deployment. The function of the console is the same in both of these deployments.

### Starting the Discovery Management Console

The Discovery Management Console is a Java-based user interface you start from a Web browser.

Make sure your browser is configured to use a supported Java™ runtime environment, and that your computer meets all TADDM client hardware and software requirements. For more information, see the *TADDM Installation Guide*.

To start the Discovery Management Console, complete the following steps:

1. Open a web browser, and type the URL and port number of the system where you installed the TADDM server. The default port number is 9430.  
For example:

```
http://system.company.com:9430
```

The TADDM Launch Page is displayed. Make sure that all services in the Administrator Console have been started.

2. Optional: To use an SSL connection, complete the following steps:
  - a. Under the Discovery Management Console heading, select **Show SSL Options**.
  - b. Click **Download Trust Store** to download the truststore and select a directory in which to save the truststore file.
  - c. In the input box to the right of the **Download Trust Store** link, enter the name of the directory that contains the truststore file.
3. Click **Start Discovery Management Console**. The **File Download** window is displayed.
4. In the **File Download** window, click **Open**. The **Login** window is displayed.
5. In the **Username** field, type the user name for connecting to the TADDM server. Use either a user account that was created during installation, or the default administrator account. The default administrator user name is `administrator`.
6. In the **Password** field, type the password for the user name that you entered. The password for the default administrator user name is `collation`.
7. In the **Server** field, type the fully-qualified server name to access. The **Server** field is prefilled with the default server name.
8. In the **Port** field, type the port number for the server. The **Port** field is prefilled with the default port number.
9. Optional: Select **Establish a secure (SSL) session** to encrypt all data, including your user name and password, before transmitting over the network. To use SSL, you must have saved the truststore for the server when you installed the Discovery Management Console client.
10. Click **Login**. The Discovery Management Console client window is displayed.

**Important:** For more information on logging into the Discovery Management Console with an SSL connection, see the *TADDM Troubleshooting Guide*.

## Discovery scopes

You can use a discovery scope to identify the devices, computer systems, and other components in your infrastructure that you want the server to access. You must configure at least one scope before running a discovery.

You can specify the discovery scope sets using IP addresses, ranges of IP addresses, or subnets to define the boundary of the networks that can be accessed during discovery. A scope can be as small as a single IP address, or as large as a range of IP addresses or a class C network. To scan subnet ranges greater than a class C network, see [Loading a discovery scope from a file](#) for more information. You can also exclude specific devices from the scope.

When there is a firewall between the server and the systems that you want discovered in another area of your network, configure the firewall to allow access on the SSH port (port 22) and then set up an anchor. See [“Anchors and gateways”](#) on page 41 for more information. The following table lists and describes the information that is displayed for a discovery scope sets in the **Scope** pane, in the **Scope Sets** tab:

Discovery scope information	Description
Method	Specifies whether to include or exclude the IP address, IP address range, or subnet.
Type	<p>The type of address specified, from among the following options:</p> <p><b>Subnet</b> An IP subnet, for example, 255.255.255.0.</p> <p><b>Range</b> IP address range, for example, 1.2.3.4 - 1.2.3.10</p> <p><b>Host</b> IP address, for example, 1.2.3.4.</p>
Value	The actual IP address, IP address range, or subnet.
Description	A user-supplied description or host name of the discovery scope.

**Note:** A scope, or a scope group, is not an identifier, but a collection of individual IP addresses. Therefore, if you restrict configuration, for example Access Entry, or Discovery Profile, to a scope, or a scope group, it applies to all IP addresses included in that scope, or scope group. It also means that when a given IP address is included in many scopes or scope groups, and you restrict configuration for only one of them, the restriction for the given IP always applies, no matter which scope, or scope group, is used for discovery.

## Configuring a scope

You can use the Discovery Management Console to configure a scope set.

**Important:** Creating very large scopes can lead to performance issues, including a server crash.

To configure a scope set and scope, complete the following steps from the Discovery Management Console:

1. On the menu bar, click **Discovery > Scope**.  
The **Scope** pane is displayed.
2. To define a new discovery scope set, click **Add Set**.  
The **Scope Set Name** window is displayed.
3. In the **Name** field, type the name for the new scope set.

**Important:** The scope set names cannot contain the following characters:

- .
- /

**Note:** If you are managing multiple domains with a synchronization server, make sure each scope set name is unique within all domains managed by the same server. Using the same scope set name in more than one domain can cause problems when generating reports.

4. Click **OK**. The new scope set is displayed in the Scope Sets list.
5. To add the scope and contents to the scope set, select the scope set that you just created and click **Add**.

The **Add Scope** window is displayed.

6. To add the settings for the scope, complete one of the following steps:
  - Select **Subnet** from the **IP Type** list and type the IP address of the subnet mask in the IP Address field. This must be a unique value within the scope set.
  - Select **Range** from the **IP Type** list and type the start and the end IP addresses in the IP Addresses field. This must be a unique value within the scope set.
  - Select **Host** from the **IP Type** list and type the IP address of the host in the IP Address field, or type the host name in the **Hostname** field. This must be a unique value that exists within the scope set.

**Important:** If both the IP and host name are defined and do not correspond to each other, the IP takes precedence. The host name is only treated as a "description".

7. To exclude devices and hosts from your scope, click **Add Exclusion** and complete one of the following steps:
  - From the **IP Type** list, select **Subnet** and type the IP address of the subnet in the **IP Address** field.
  - From the **IP Type** list, select **Range** and type the start and end IP addresses in the **IP Address** field.
  - From the **IP Type** list, select **Host** and type the start and end IP addresses in the **IP Address** field.
8. To save the scope, click **OK**. The new scope is displayed in the list.

## Configuring a scope group

You can use the Discovery Management Console to configure a scope group.

1. To add the group of scope sets, complete the following steps:
  - a) In the Functions pane, click **Discovery > Scope** and select **Scope Groups** tab.
  - b) To create a new empty scope group, click **Add Set**. The **Scope Group Name** window is displayed.
  - c) In the **Name** field, type MyGroup as the name for the new scope group.
  - d) Click **OK**. The name MyGroup is displayed in the Scope Groups list.
2. To add existing scope sets to the scope group, complete the following steps:
  - a) From the list of Scope Groups in the **Scope Groups** tab, select **MyGroup** and click **Add**. The **Add scope sets to Group** window is displayed.
  - b) Select scope sets that you want to add to group.
  - c) Click **Add**.

The new scope sets are displayed on the list.

## Changing a scope

You can use the Discovery Management Console to change an existing discovery scope.

**Important:** Creating very large scopes can lead to performance issues, including a server crash.

To change an existing discovery scope, complete the following steps from the Discovery Management Console:

1. On the menu bar, click **Discovery > Scope**.  
The **Scope** pane is displayed.

2. From the **Scope Sets** list, select a scope set.  
The list of scopes for that scope set are listed to the right.
3. From the list of scopes, select a scope and click **Edit**.  
The **Edit Scope** window is displayed.
4. To change the settings for the scope, complete one of the following steps:
  - To change a subnet, type the IP address of the subnet in the **IP Address** field. This must be a unique value within the scope set. Continue to Step 5.
  - To change a range of devices, type the start and the end IP addresses in the **IP Address** field. This must be a unique value within the scope set. Continue to Step 5.
  - To change a specific device, type the IP address or the fully qualified host name in the **Hostname** field. This must be a unique value within the scope set. Continue to Step 6.
5. To exclude devices and hosts from the discovery scope, click **Add Exclusion** and complete one of the following steps:
  - From the **IP Type** list, select **Subnet** and type the IP address of the subnet in the **IP Address** field.
  - From the **IP Type** list, select **Range** and type the start and end IP addresses in the **IP Address** field.
6. To save the scope, click **OK**. The new changes are applied to the scope.

## Changing a scope group

You can use the Discovery Management Console to change an existing discovery scope group.

To change an existing discovery scope group, complete the following steps from the Discovery Management Console:

1. On the menu bar, click **Discovery > Scope** and select the **Scope Groups** tab.
2. From the list of scope groups, select the group that you want to edit.
  - To delete a scope set from the scope group, select the scope sets that you want to delete and click **Delete from a group**.
  - To add a scope set to a scope group, click **Add to group**. The **Add Scope** window opens. Select scope sets that you want to add to the scope group and click **Add**.

## Deleting a scope

You can use the Discovery Management Console to delete a scope.

To delete a scope, complete the following steps from the Discovery Management Console:

1. On the menu bar, click **Discovery > Scope**  
The **Scope** pane is displayed.
2. From the **Scope Sets** list, select the scope set that contains the scope that you want to delete.  
The scopes for that scope set are listed to the right.
3. From the list of scopes, select a scope and click **Delete Set**.  
A message window is displayed.
4. To delete the scope, click **Yes**.  
The scope is deleted from the scope set.

## Deleting a scope set

You can use the Discovery Management Console to delete a scope set.

To delete a scope set, complete the following steps from the Discovery Management Console:

1. On the menu bar, click **Discovery > Scope**  
The **Scope** pane is displayed.
2. From the **Scope Sets** list, select the scope set that you want to delete, and click **Delete Set**.



- A message window is displayed.
3. To delete the scope set, click **Yes**.
- The scope set is deleted.

## Deleting a scope group

You can use the Discovery Management Console to delete a scope group

To delete a scope group, complete the following steps from the Discovery Management Console:

1. On the menu bar, click **Discovery > Scope**. The Scope pane is displayed.
2. Select the **Scope Group** tab.
3. From the Scope Groups list, select the scope group that contains the scope that you want to delete. The scopes sets for that scope group are listed to the right.
4. From the list of scopes, select a scope and click **Delete Group**. A message window is displayed.
5. To delete the scope group, click **Yes**. The scope group is deleted.

## Access lists

The access list is a collection of all user names, passwords, and Simple Network Management Protocol (SNMP) community strings that the server uses when accessing the configuration items in your infrastructure. You must set up this list for the Configuration Items that you want to discover. When using the Stack Scan sensor for credential-less discovery, an access list is not required.

User names, passwords, and community strings if needed, are categorized by each type of device or software application, and optionally restricted by scope. For example, all user names and passwords for all computer systems are stored as a group, and all user names and passwords for all databases are stored as another group.

When accessing a device, the server sequentially uses each user name and password (or community string) in the group across a particular scope (IP address per subnet) until the device allows the server permission to access it. For example, when accessing a computer system, the server uses the first user name and password specified in the access list for computer systems. If the user name and password are incorrect for a particular computer system, the server automatically uses the next user name and password that is specified in the access list for a computer system.

Because you enter a list of user names and passwords (or community strings) for each type of configuration item, you do not need to specify a user name and password for a particular configuration item. When you specify all user names and passwords for each type of device, define the scope for each user name and password pair. The server automatically tries each user name and password until the correct combination is found. The access list that you create is used by the Discovery Management Console and is encrypted and stored in the database.

If the device you are discovering is a network device capable of being managed through the SNMP protocol, enter an SNMP community string in the Community field. If you are using SNMP for a Cisco device, you must select the SNMP network element and enter an SNMP community string in the Community field for the Cisco device.

For each Computer System entry in the access list, you have the option to specify one of the following authentication types:

- default
- password
- public key infrastructure (PKI)

If you select default authentication, SSH key-based authentication is attempted first, using the password for the key passphrase, if required. If key-based authentication does not succeed, then login name and password authentication is attempted. If password authentication type is selected, only password authentication is attempted. Similarly, if PKI is selected, only key-based authentication is attempted. It is recommended that you set the authentication type for the new access list entry being added if you know

the type. If you do not know the authentication type, the default behavior can lead to a lot of invalid login attempts that can sometimes result in the user being locked out of the account.

In cases when your system administrator has set up SSH with the login and password authentication method, start the Discovery Management Console with the Establish a Secure (SSL) Session option enabled before you set up the access list. This option encrypts all data including access list user names and passwords before the data is transmitted between the Discovery Management Console and the server.

## Adding a new access list entry

You can add a new access list entry using the Discovery Management Console. The steps for adding a new access list entry vary, based on the component type that you want to add. Use the **Discovery > Access List** to add a new access list entry. You can also programmatically add new access list entries using the Java API.

If you want to programmatically add new access list entries, or if you have a vendor supplied application and want to manage identities or change the password, use the Java API to carry out this task. For the Java API methods, see the *Managing access lists* topic in the *TADDM SDK Developer's Guide*.

To add a new access list entry using the Discovery Management Console, complete the following steps:

1. On the menu bar, click **Discovery > Access List**.

The **Access List** pane is displayed.

2. To add a new entry into the access list, click **Add**.

The **Access Details** notebook is displayed.

3. From the **Component Type** list, select the component type that you want to discover.

4. For all component types other than Network Element (SNMP), complete the following steps:

- a) In the **Name** field, type the name of the access list entry.

- b) In the **User name** field, type the user name to log in to the component that you want to discover.

When specifying a Windows domain user account, the domain name and user name must be separated by a backslash (\) as shown in the following example: DOMAIN\username.

- c) In the **Password** field, type the password to log in to the component that you want to discover.

- d) In the **Confirm Password** field, retype the password to log in to the component that you want to discover.

5. Click **OK** to save your information.

The **Access List** pane is displayed with the new information.

6. Additional steps can be required based on the component type that you select. The following table identifies the component types and the additional fields and lists that you are required to complete for the access list entry.

Table 2. Required component types, fields, and lists for access list entry	
Component Types	Fields and Lists
Application Server, Database, Messaging Servers	<b>Name</b> Name to identify the device in the access list. <b>User name</b> User name to access the server. <b>Password</b> Password to access the server. <b>Vendor</b> The vendor of the server or database.

Table 2. Required component types, fields, and lists for access list entry (continued)	
Component Types	Fields and Lists
CSM Server	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>Password</b> Password to access the server.</p> <p><b>User name</b> User name to access the server.</p>
Cisco Device	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the device.</p> <p><b>Password</b> The password for the Cisco device, if you are using Telnet protocol, SSH1 or SSH2.</p> <p><b>Enable Password</b> The Enable password for the Cisco device, if you are using Telnet protocol, SSH1 or SSH2.</p> <p><b>Confirm Enable Password</b> The Enable password for the Cisco device, if you are using Telnet protocol, SSH1 or SSH2.</p> <p>The Cisco IOS sensor requires the SNMP sensor to be established and working against the device. If your Cisco IOS sensor is using a Telnet protocol and does not prompt for a user name, type default in the <b>User name</b> field.</p>
Cisco Works	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the server.</p> <p><b>Password</b> Password to access the server.</p>
Computer System, Computer System (Windows)	<p><b>Authentication Type</b> The type of authentication for the computer system.</p> <p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the computer system.</p> <p><b>Password</b> Password to access the computer system.</p>

Table 2. Required component types, fields, and lists for access list entry (continued)	
Component Types	Fields and Lists
Computing Center Management System (CCMS)	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the SAP CCMS server.</p> <p><b>Client ID</b> The client ID of the SAP CCMS server.</p> <p><b>Password</b> Password to access the SAP CCMS server.</p>
High Availability Solutions	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the Veritas Cluster server.</p> <p><b>Password</b> Password to access the Veritas Cluster server.</p>
IBM Tivoli Monitoring	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the Tivoli Enterprise Portal Server.</p> <p><b>Password</b> Password to access the Tivoli Enterprise Portal Server.</p>
LDAP Service	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the LDAP server.</p> <p><b>Password</b> Password to access the LDAP server.</p>
Network Element (SNMP)	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>Community String</b> The community string for the network device.</p> <p><b>Confirm Community String</b> The community string for the network device.</p> <p>The SNMP Network element must be configured to answer queries from the TADDM server IP address.</p>

Table 2. Required component types, fields, and lists for access list entry (continued)	
Component Types	Fields and Lists
Network Element (SNMPV3)	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the device.</p> <p><b>Password</b> Password to access the device.</p> <p><b>Private Password</b> The password used if data encryption is set for SNMP.</p> <p><b>Authentication Protocol</b> The type of authentication protocol used by SNMP.</p>
SysImager Server	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the SysImager server.</p> <p><b>Password</b> Password to access the SysImager server.</p>
System Landscape Directory Server	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the System Landscape Directory Server.</p> <p><b>Password</b> Password to access the System Landscape Directory Server.</p>

7. To configure the scope limitations, click the **Scope Limitations** tab.  
The **Scope Limitations** page is displayed.
8. On the **Scope Limitations** page, complete one of the following steps:
  - To use the access information across all components of the entire discovery scope, click **Entire scope**.
  - To restrict the application of specific access information to certain systems, click **Limit to selected scope** or **Limit to selected scope groups**. Then select the scope set or scope group to which you want to restrict access. The access list entry is only used when discovering the selected scope. When discovering a different scope set or scope group, the access list entry is not used. This method prevents invalid login attempts that can result in the user being locked out of the account.
9. To save the new access list entry, click **OK**.

## Changing an access list entry

You can change an existing access list entry using the Discovery Management Console. The steps for changing an access list entry depend on the component type that you want to add. Use the **Discovery > Access List** to change an existing access list entry. You can also programmatically change existing access list entries using the Java API.

If you want to programmatically manage your access list entries, or if you have a vendor supplied application and want to manage identities or change the password, for example, you can use the Java API

to do this. For the Java API methods, see the *Managing access lists* topic in the *TADDM SDK Developer's Guide*.

To change an access list entry using the Discovery Management Console, complete the following steps:

1. From the menu bar, click **Discovery > Access List**.  
The **Access List** pane is displayed.
2. From the list, select the entry that you want to change and click **Edit**.  
The **Access Details** notebook is displayed, with the **Component Type**, **Name**, and **User name** fields disabled. You cannot change these settings.
3. If you want to change the password information, click **Change** and complete the following steps:
  - a) In the **Password** field, type the password to log into the component that you want to discover.
  - b) In the **Confirm Password** field, retype the password to log into the component that you want to discover.
4. To configure the scope limitations, click the **Scope Limitations** tab.  
The **Scope Limitations** page is displayed.
5. On the **Scope Limitations** page, complete one of the following steps:
  - To use the access information across all components of the entire discovery scope, click **Entire scope**.
  - To restrict the application of specific access information to certain systems, click **Limit to selected scope** or **Limit to selected scope groups**. Then select the scope set or scope group to which you want to restrict access. The access list entry is only used when discovering the selected scope. When discovering a different scope set or scope group, the access list entry is not used. This prevents invalid login attempts that can result in the user being locked out of the account.
6. To save the new access list entry, click **OK**.

## Moving an access list between servers

You can move an access list from one TADDM server to another TADDM server.

You cannot move an access list either to or from a primary storage server or a secondary storage server.

To move an access list from one TADDM server to another, complete the following steps:

1. Open a command-line window on the TADDM server that you want to move the access list from.
2. From the `$COLLATION_HOME/bin` directory, use the following command to write the access list to an encrypted file:

**On Linux, AIX, and Linux on System z® operating systems:**

```
authconfig.sh -u $username -p $password -d -f $filename
```

**On Windows operating system:**

```
authconfig.bat -u username -p password -d -f filename
```

3. Copy the encrypted file created by the `authconfig` command and the `TADDMSec.properties` file from `$COLLATION_HOME/etc/` directory to the TADDM server that you want to move to (destination server). Ensure that you do not overwrite the existing `TADDMSec.properties` file on the destination server.
4. On the TADDM server that you have moved the files to, from the `$COLLATION_HOME/bin` directory, use the following command:

**On Linux, AIX, and Linux on System z operating systems:**

```
authconfig.sh -u $username -p $password -m  
-f $filename -k $key_filename [-o] [-e $output_filename]
```

### On Windows operating system:

```
authconfig.bat -u username -p password -m  
-f filename -k key_filename [-o] [-e output_filename]
```

where:

#### **-f filename**

This value specifies the name and location of the encrypted file that was created in step 2.

#### **-k key\_filename**

This value specifies the name and location of the TADDMSec.properties file that you copied from the source TADDM server. The TADDMSec.properties file on the destination server cannot be used when moving an access list that was encrypted by the source server.

#### **Optional: -o**

The default action when moving an access list between servers is to merge the encrypted files access list with the access list on the destination server. Then the combined list is saved to the database.

To overwrite the current access list on the destination server with the access list in the encrypted file, add the -o option.

#### **Optional: -e output\_filename**

If you do not want to immediately write the access list from the encrypted file to the database on the destination server, add the -e option. The -e option re-encrypts the access list to an output file using the encryption key of the destination server. You must specify a name for the re-encrypted file and a location for it on the destination server. This option does not save any access list entries from the encrypted file to the database on the destination server.

If you have selected the -e option when moving access lists between servers, you can move the re-encrypted file at a later time to the database of the destination server. When moving the re-encrypted file to the database, use the following command, where filename is the name and location of the re-encrypted file:

### On Linux, AIX, and Linux on System z operating systems:

```
authconfig.sh -u $username -p $password -m -f $filename [-o]
```

### On Windows operating system:

```
authconfig.bat -u username -p password -m -f filename [-o]
```

The -o is optional and behaves exactly as outlined in the preceding procedure. Ensure that you omit the -k option. The TADDMSec.properties file on the remote server cannot be used to move the re-encrypted file to the database.

## Running discoveries

After you set up an initial scope for the discovery and establish an access list for your computing systems, you are ready to run a basic discovery. You can also run a non-admin Windows discovery for which you do not need to provide the administrator account.

### Running a basic discovery

After you set up an initial scope for the discovery and establish an access list for your computing systems, you are ready to run a basic discovery.

**Important:** Running a discovery against a very large scope can lead to performance issues, including server crash.

To run a discovery, complete the following steps from the Discovery Management Console:

1. From the menu bar, click **Discovery > Overview**. The **Overview** pane is displayed.

2. To start the discovery, click **Run Discovery**. The **Run Discovery** window is displayed.
3. In the Run Discovery window, select **Selected Scope Elements** from the **Scope** menu, and then select from the tree the scopes to include in the discovery. You can run discovery against scope elements, scope sets and groups of scopes.
4. From the **Profile** list, select the discovery profile to use during the discovery run. See [“Using discovery profiles” on page 50](#) for more information about discovery profiles.
5. To run the discovery, click **OK**.

After you start a discovery, and while the discovery is running, you can view the **Discovery Overview** to view details of your discovery.

See [“Setting up a discovery” on page 168](#) for a scenario-driven approach to discovery.

## Rediscovering configuration items

You can use the **Rediscover** option to refresh a configuration item (CI) that was already discovered by a credentialed discovery.

The **Rediscover** option is available in Data Management Portal. Use this option to rediscover the CI without going through the full discovery process.

Before you can rediscover configuration items, you must edit the following property in the `collation.properties` file:

### **com.collation.rediscoveryEnabled=true**

Valid values are `true` and `false`. The default is `false`. Change the value to `true` to enable the rediscovery function.

**Fix Pack 5** If any CI is to be rediscovered by invoking the main sensor only due to which the CI is created, the following property should be set to `true`:

```
com.collation.isRediscoveryViaMainSensorOnly=true
```

Valid values of this property are `true` and `false`. The default is `false`.

The **Rediscover** option uses information stored by the prior full discovery to seed the rediscovery. Aged seeds might provide unpredictable results. In the following examples, the changes after a full discovery might cause a seed to become invalid, and the rediscovery might fail or obtain incomplete data:

- The target information that the sensor uses changed (IP address, port binding, and so on).
- The underlying data model changed. This situation is typical with releases and maintenance, such as a new release, fix pack or an interim fix.
- The sensor changes significantly, which affects the seed information that is stored.

Plan a full discovery of all CIs after major maintenance applications, such as installing an interim fix, fix pack or new release, instead of using the **Rediscover** option. By doing so, you ensure that the seeds are maintained at an adequate level for the rediscovery to function correctly.

### **Important:**

- Rediscovery is not a long-term method to keep CIs up to date.
- For the rediscovery of a custom collection, the process is different. A custom collection itself cannot be rediscovered but the elements that belong to it can be. When you select a custom collection to rediscovery, the elements that belong to it and can be rediscovered are automatically rediscovered.
- **Fix Pack 3** In TADDM 7.3.0.3, and later, you can rediscover only top level objects.
  1. In Data Management Portal, go to the lower left **Discovered Components** pane.
  2. Select the CIs that you want to rediscover.
  3. Click **Actions > Rediscover**.



4. Rediscovery starts only the last sensor used to discover the particular CI. It does not start further downstream sensors for a deeper discovery. To run a full discovery, use **Run Discovery** in Product Console.

After you complete a rediscovery and you view the discovery history in the Product Console, you see that the sensors ran, but the profile and the scope fields are blank. The rediscovery creates a dynamic profile whenever you run a rediscovery.

## Managing discoveries

You can use the Discovery Management Console to manage the discovery process.

### Capturing discovery results on the discovery server

In a streaming server deployment, discovery data flows from the discovery server to the primary storage server. This data is processed and stored in the database on the primary storage server. You can alternatively capture and store this information in the file system on the discovery server. This information can be analyzed and transferred to the storage server for processing and storage at a later time.

To capture the discovery information on the discovery server, complete the following steps:

1. In the `$COLLATION_HOME/var` directory, create a directory named `topo` to store the discovery information. When a discovery is run, the results are stored in a unique directory created in the `$COLLATION_HOME/var/topo` directory. For example, the `$COLLATION_HOME/var/topo/74i9x86th` directory.
2. To store the information in the database, complete the following steps:
  - a) Copy the generated directory to the storage server. The same TADDM version must be installed on the storage server and discovery server.
  - b) Run the following command:

```
api.sh -u username -p password find "IMPORT data_directory"
```

where `data_directory` is the directory from which the data is imported.

For example,

```
api.sh -u administrator -p collation find "IMPORT /tmp/74i9x86th"
```

The `api.sh` script is located in the `$COLLATION_HOME/sdk/bin` directory.

The discovery server cannot persist the information to the database during a discovery until the `$COLLATION_HOME/var/topo` directory is deleted.

In a domain server deployment, the domain server has its own database. For a stand-alone domain server, the information from a discovery can be captured on the server by using the previous steps.

3. To list the contents of stored objects, run the following command:

```
api.sh -u username -p password find "LIST data_directory"
```

where `data_directory` is the directory containing the data to be analyzed. Typically this is the same directory as the one to which you exported the discovery data, for example,

```
api.sh -u administrator -p collation find "LIST /tmp/74i9x86th"
```

The contents of the stored objects is listed in `$COLLATION_HOME/log/services/ApiServer.log`.

There is a limit to the amount of data logged in each message. For large objects, the value of the **com.collation.log.msg.size** property must be increased above the default value of 100000.

## Loading a discovery scope from a file

You can use the **loadscope** command to manage and load the discovery scope from a file.

**Important:** Creating large scopes can lead to performance issues, including a server crash.

The following example shows the format of the **loadscope** command:

```
loadscope.jy [-d] [-q] [-C] -u username -p password clearAll |  
(clearScopename) | (clearScopeSetName) |  
([-s ScopeSetName | -g ScopeGroupName] load [scopefile])
```

The `loadscope.jy` script is in the `$COLLATION_HOME/bin` directory.

The following list describes the **loadscope** command options:

### -d

Turns on verbose debug logging.

### -q

Loads the scope without synchronization.

You can use this option when you load multiple scopes. Ensure that you do not use the `-q` flag with the final one so that synchronization can then take place.

**Important:** **Fix Pack 3** In TADDM 7.3.0.3, and later, this option is ignored. As the performance of scope synchronization improved significantly, this option is no longer needed.

### -C

This parameter makes the `loadscope.jy` file delete the scope. However, it does not delete the `ScopeElements` assigned to the scope that is later removed by a Topology Builder agent.

### -u username

The user name to access the TADDM server. This parameter is mandatory for load operations.

### -p password

The password for the user name. This parameter is mandatory for load operations.

### clearAll

Deletes all scope sets and scope group.

### clearScope

Deletes scope set or scope group.

### clearScopeSet

**Important:** Deprecated.

Deletes scope set or scope group.

### -s ScopeSet or -g ScopeGroup

**ScopeSet** is the scope set name used for loading the scope elements. **ScopeGroup** is the scope group name used for loading the scope sets. This parameter is mandatory for load operations.

**Important:** The scope set names cannot contain the following characters:

- ' (single quote)
- . (dot)
- / (forward slash)

### load

Loads the scope elements to the system, replacing existing elements with new elements.

### scopefile

The file that contains the following elements:

- The scope elements in case you want to load a scope set by using `-s ScopeSet` parameter.
- The scope sets in case you want to load a scope group by using `-g ScopeGroup` parameter.

This parameter is mandatory for load operations.

## Loading a scope set from file

The following example shows how to load a discovery file by using the **loadscope** command:

```
% loadscope.jy -u administrator -p cmdb -s Windows load /tmp/scopefile
```

A scope file consists of entries in the following format:

```
scope, [exclude_scope:exclude_scope...],[description]
```

A scope file can contain any number of scopes, by using any combination of the following scope types:

- Subnet scopes (for example, 1.2.3.4/255.255.255.0)
- Address scopes (for example, 1.2.3.4)
- Range scopes (for example, 1.2.3.4-5.6.7.8)

**Important:** The following are details about the scope file format:

- Only IP addresses are valid in the scope file. Host names cannot be used.
- Each scope element exists on a separate line.
- Address scopes must not include exclusions.
- The ampersand character (&) is not allowed in the **[description]** parameter.
- Entries are ignored if they are not valid.
- You can insert comment lines prefixed with the number sign (#).

The following sample text is a sample scope file:

```
# This is a comment
10.10.10.10,,
10.10.10.20,,
10.10.10.30,,
10.10.10.0/255.255.255.0,10.10.10.2:10.10.10.3,
10.10.10.2-10.10.10.9,10.10.10.4:10.10.10.5,
10.10.10.88,,
10.10.10.999,,
```

## Loading a scope group from file

A scope file consists of entries in the following format, which describes a single scope group:

```
scopeSetName1
scopeSetName2
...
```

```
scopeSetNameN
```

where scopeSetNameN is the name of an existing scope set that is to be added into the group.

**Important:** Each scope set name is placed on a separate line. You can insert comment lines with the number sign (#) at the beginning.

Use the following commands to load and delete scope sets and scope groups:

- Loading a scope set:

```
loadscope.jy -u <username> -p <password> -s <ScopeSet> load <scopefile>
```

- Loading a scope group:

```
loadscope.jy -u <username> -p <password> -g <ScopeGroup> load <scopefile>
```

- Deleting a scope set or a scope group:

```
loadscope.jy -u <username> -p <password> clearScope <name>
```

- Deleting all scope sets and scope groups:

```
loadscope.jy -u <username> -p <password> clearAll
```

## Exporting scopes for use on another TADDM server

Use the **api.sh** command to export scopes.

To export scope sets and scope groups in an XML format, run the following command:

```
api.sh -u -p find --depth=5 Scope
```

Locate the **api.sh** script in the `$COLLATION_HOME/sdk/bin` directory.

**Restriction:** Scopes exported with the **api.sh** command cannot be imported into another TADDM server. Use the **datamover.sh|bat** command to move scopes between TADDM servers.

To maintain data integrity, you must move the data between the same versions of TADDM servers.

## Creating and managing custom server templates

You can create custom servers to discover and categorize servers that are not, by default, supported by TADDM. This is an advanced technique for configuring TADDM to discover servers that it does not know about by default.

Your infrastructure might contain software applications and server types, such as custom Java servers, that are not automatically categorized by TADDM. Any server process with a TCP listening port that is not recognized is categorized into an Unknown Server category. Unknown servers are not displayed in the topology and cannot take advantage of most of the functions. You do, however, get basic information such as the name and runtime data about the unknown server.

You can define a custom server to create a template that sets up the membership rules for the custom server. During a discovery, any unknown server is automatically categorized as a custom server of this type if the runtime information matches the criteria that you defined in the template. Any configuration files used by the custom server are also automatically captured if specified in the templates.

A predefined custom server template, called "Ignore all unmatched processes", ignores any processes that are not matched by another template. A performance improvement is gained by using this template, but if you want to search for unknown servers using the Unknown Processes functionality, you must ensure that this template is not enabled. By default, the "Ignore all unmatched processes" template is not enabled.

Custom servers are displayed in the topology, and you can view details about them. Although these details are not as complete as those provided for supported servers, defining custom servers allows all components in your infrastructure to participate in the topology and comparisons. You can manage custom servers in the **Custom Servers** window.

**Fix Pack 8** When we define and enable a Custom Server, then basic information for example name, runtime data, etc, can be captured along with the configuration files defined in the custom server template. The Object created is of the type defined in the custom server template definition. Some examples to configure and capture configuration files are also given at . We can also extend a custom server to collect additional information, refer to the 'Extending custom server and computer system templates' topic in the *Using Guide*.

For example, in order to discover and categorize java servers if not discovered by default in TADDM, a custom server template can be defined with:

1. Type as an AppServer
2. The Identifying Criteria as "Program Name contains Java"

With the above configuration, a CustomAppServerSensor will be invoked corresponding to each java process running on the target (`ps -eaf | grep -i java`)

**Fix Pack 2** In TADDM 7.3.0.2, and later, the `hierarchyType` attribute is set for each custom server template. It is used to define in more detail the source and target objects of the relationships to traverse

in the traversal section of the grouping pattern configuration. The value of this attribute is created based on the templates names. All spaces are removed and the first letters of separate words are set to upper case. For example, the IBM Tivoli Enterprise Console template has `hierarchyType` attribute set to `IBMTivoliEnterpriseConsole`, and the CA iTechnology iGateway template has the `hierarchyType` set to `CAITechnologyIGateway`.

**Fix Pack 3** In TADDM 7.3.0.3, and later, you can enable the creation of placeholders, which is useful in creating custom server templates. For details, see the *Configuring for discovery of placeholders* topic in the *TADDM Administrator's Guide*.

## Adding custom servers

A custom server template contains descriptive criteria that is used to assign unknown server processes to the custom server. You specify this criteria when defining the template for the custom server in the Discovery Management Console.

The following information associated with running processes is parsed to match the process to a particular custom server:

### Program name

The name of the executable program.

### Window Service name

The name of a Window operating system service.

### Argument

The arguments passed to the program.

### Environment

The environment variables set for the program.

### Port

The TCP port number on which the process is listening.

The custom server general information and criteria details include the name, the type of server, and identifying criteria for the custom server. To view details about that unknown server, double-click an unknown server in the Topology, and click the **Runtime** tab.

You can then use this information to create a search criteria for a custom server using the **General Information** and **Criteria** tab of the **Custom Server Details** window.

To add a custom server, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Custom Servers**.  
The **Custom Servers** pane is displayed.
2. In the **Custom Servers** pane, click **Add**.  
The **Custom Server Details** notebook is displayed.
3. In the **Name** field, type the name of the custom server.
4. From the **Type** list, select the type of custom server that you are adding.
5. Under Action, complete one of the following steps:
  - Click **Discover** if you want to discover all instances of the server.
  - Click **Ignore** if you want to suppress discovery of all instances of the server.
6. To enable the custom server definition, click **Enabled**.
7. To select an icon to associate with the custom server, click **Browse** and select the icon that you want to use.
8. Under Identifying Criteria, complete one of the following steps:
  - To match all of the identifying criteria, click **All Criteria**.
  - To match any of the identifying criteria, click **Any Criteria**.
9. Complete the following steps to define the criteria for the custom server:
  - a) From the first list, select the criteria type.

- b) From the second list, select the operator.
- c) In the field provided, type the text argument for the criteria type and operator.
10. To remove the identifying criteria, click **Remove**.
11. To define new criteria, click **Add Criterion**.
12. To add configuration files, click the **Config Files** tab.  
The **Config Files** page is displayed.
13. On the **Config Files** page, click **Add**.  
The **Search Path for Capture File** window is displayed.
14. From the **Type** list, select one of the file types to capture:
  - Config File
  - Software Module
  - Application Descriptor Directory/File
15. From the **Search Path** list, select one of the following search paths for the configuration file:
  - /**  
The root of the file system.
  - \$PWD**  
The current working directory of the running program.
  - \$Home**  
The home directory of the user ID of the running program.
  - C:**  
A directory on your local computer.
  - %ProgramFiles%**  
The program files directory.
  - %SystemRoot%**  
The system root directory

Type the path and file name of the configuration file in the text box, or type \* (asterisk) to specify all files in the selected directory.
16. To capture the contents of the configuration file, click **Capture file contents** and optionally specify the maximum number of bytes of the captured configuration file.
17. To recurse through the directory structure to search for the specified file, select **Recurse Directory Search** (if you use TADDM 7.3.0.3, or later), or **Recurse Directory Content** (if you use TADDM 7.3.0.2, or earlier).
18. To save the settings for your custom server, click **OK**.

### ***Editing a custom server***

You can use the Discovery Management Console to edit a custom server.

To edit a custom server, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Custom Servers**.  
The **Custom Servers** pane is displayed.
2. In the **Custom Servers** pane, click **Edit**.  
The **Custom Server Details** notebook is displayed, with the **Name** and **Type** fields disabled. These fields cannot be changed.
3. To change the other fields in the **Custom Server Details** notebook, see [“Adding custom servers” on page 17](#).
4. To refresh the information about the custom server you just changed, run another discovery.  
To improve the speed of the discovery process, limit the active scope of the discovery to the new component.

### ***Copying a custom server***

You can create a new custom server based on an existing one. This is done by copying a server listed in the **Custom Servers** pane and issuing it a unique name.

To copy a custom server, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Custom Servers**.  
The **Custom Servers** pane is displayed.
2. In the **Custom Servers** pane, select the custom server that you want to copy and click **Copy**.  
The **Set Name** window is displayed.
3. In the **Name** field, type the name for the new custom server.
4. To save the new custom server, click **OK**.

### ***Deleting a custom server***

You can use the Discovery Management Console to delete a custom server.

To delete a custom server, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Custom Servers**.  
The **Custom Servers** pane is displayed.
2. In the **Custom Servers** pane, select the custom server that you want to delete and click **Delete**.  
A message window is displayed.
3. To delete the custom server, click **Yes** in the message window.
4. To confirm the deletion, ensure that the custom server is not listed in the **Custom Servers** pane.

### ***Repositioning custom server entries***

You can change the order in which custom servers are listed in the **Custom Servers** pane. The list order is important because template matching is applied from top to bottom in the custom server list and stops at the first match. For example, a more generic template matches all servers of a specific type and a more specific template matches only servers that have a specific string argument. After a server is matched to a server category, the custom server is removed from the unknown server list. A server cannot be a member of more than one category at the same time, even if the server matches criteria from several custom servers in the list. Changing the order of the list can cause the server process to match to a different custom server.

To reposition entries in the **Custom Servers** pane, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Custom Servers**.  
The **Custom Servers** pane is displayed.
2. In the **Custom Servers** pane, select the custom server that you want to reposition and complete one of the following steps:
  - To move the server up in the entry list, click **Move Up**.
  - To move the server down in the entry list, click **Move Down**.

### ***Enabling the ignore template***

You can enable the predefined "Ignore all unmatched processes" template that ignores any processes that are not matched by another template.

Enabling this template results in unknown server patterns being ignored. Before you enable it, ensure that you do not want to identify unknown servers using the Unknown Processes functionality.

To enable the template that ignores all unmatched processes, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Custom Servers**.  
The **Custom Servers** pane is displayed.

2. In the **Custom Servers** pane, select the "Ignore all unmatched processes" custom server template. Click **Edit**.  
The **Custom Server Details** window is displayed.
3. Click **Enabled**.
4. Click **OK**.

### Predefined templates

TADDM includes many predefined custom server templates that are ready for immediate use. You can use them in your own environment to discover additional software.

#### Prerequisites

Before you start using custom server templates, make sure that the `buildforge.exe` and `buildforge;NODEM` values are added to the `forcedServerList` property in the `collation.properties` file in the following way:

```
com.collation.platform.os.WindowsOs.forcedServerList=w3wp;nserver;amqzma0;
bipservice;buildforge.exe
com.collation.platform.os.UnixOs.forcedServerList=amqzma0;vxconfigd;
clstrmgr;bipbroker;libvirt;buildforge;NODEM
```

#### Fix Pack 2 New templates in TADDM 7.3.0.2

In TADDM 7.3.0.2, many templates are ready for immediate use. However, you must use these templates with caution. After you upgrade to TADDM 7.3.0.2, review the list of templates manually, and decide which of them you can use safely. Depending on your configuration, not taking this step might result in generating duplicates.

For example, in an earlier TADDM release you disabled the `JavaServer` template and you decided not to discover AppServers based on Java by using the custom templates because you used Level 3 discovery to achieve it. In such situation, if you leave the new templates enabled, duplicates might be created.

**Note:** In case of migration from TADDM 7.3.0, or 7.3.0.1, the following templates are moved to the end of the custom servers list so that the unknown servers can be first categorized as servers of the types that are specified by more specific templates listed earlier.

- IBM WebSphere Liberty Profile
- Tomcat
- JavaServer
- Ignore all unmatched processes

#### Custom server templates list

The following table contains the list of all custom server templates that are available for immediate use. It specifies whether the template is enabled by default and what object class it discovers. If an extension is created for a template, it is also listed in the table. The templates are sorted in the alphabetical order. If a template has an extension, the name of the extension is specified in the **Name** column.

Table 3. Predefined custom server templates.		
Name	Enabled by Default	Object Class Discovered
<b>Fix Pack 2</b> AppDynamics Controller	Yes	AppServer
<b>Fix Pack 3</b> Aternity Watchdog	Yes	AppServer
<b>Fix Pack 2</b> BlackBerry Enterprise Server	Yes	AppServer
BMC Patrol Agent	Yes	AppServer
BroadVision	Yes	AppServer



Table 3. Predefined custom server templates. (continued)

Name	Enabled by Default	Object Class Discovered
<b>Fix Pack 2</b> Business Objects Enterprise	Yes	AppServer
<b>Fix Pack 2</b> CA iTechnology iGateway	Yes	AppServer
<b>Fix Pack 2</b> Cisco Data Center Network Manager	Yes	AppServer
CollationProcesses	Yes	AppServer
Compaq Insight Manager Agent	Yes	AppServer
ConnectDirect	Yes	AppServer
Dell OpenManage Agents	Yes	AppServer
<b>Fix Pack 2</b> Hitachi Hi-Track Monitor	Yes	AppServer
<b>Fix Pack 2</b> HP Discovery & Dependency Mapping Inventory	Yes	AppServer
<b>Fix Pack 2</b> HP SiteScope	Yes	AppServer
<b>Fix Pack 2</b> HP System Management Homepage Software	Yes	AppServer
HTTP Server	Yes	WebServer
<b>Fix Pack 2</b> IBM Build Forge Agent	Yes	AppServer
<b>Fix Pack 2</b> IBM Build Forge Server	Yes	AppServer
<b>Fix Pack 2</b> IBM Cognos 7 PowerPlay	Yes	AppServer
<b>Fix Pack 2</b> IBM Cognos Business Intelligence	Yes	AppServer
<b>Fix Pack 2</b> IBM Cognos Impromptu	Yes	AppServer
<b>Fix Pack 2</b> IBM Communications Server	Yes	AppServer
<b>Fix Pack 2</b> IBM Content Analytics	Yes	AppServer
<b>Fix Pack 2</b> IBM Endpoint Manager for Software Use Analysis	Yes	AppServer
<b>Fix Pack 2</b> IBM Enterprise Integrator	Yes	AppServer
<b>Fix Pack 2</b> IBM InfoSphere Data Stage	Yes	AppServer
<b>Fix Pack 2</b> IBM InfoSphere Guardium	Yes	AppServer
<b>Fix Pack 2</b> IBM License Metric Tool	Yes	AppServer
<b>Fix Pack 2</b> IBM Netcool/Impact Server	Yes	AppServer
<b>Fix Pack 2</b> IBM Netcool/Impact UI Server	Yes	AppServer
<b>Fix Pack 2</b> IBM Netcool/OMNIBus	Yes	DatabaseServer

Table 3. Predefined custom server templates. (continued)



















Name	Enabled by Default	Object Class Discovered
 IBM Security AppScan Enterprise	Yes	AppServer
 IBM Security Directory Server	Yes	AppServer
IBM Tivoli Business Service Manager	Yes	AppServer
IBM Tivoli Enterprise Console	Yes	AppServer
 IBM Tivoli Enterprise Monitoring Server	Yes	AppServer
 IBM Tivoli Enterprise Portal Server	Yes	AppServer
 IBM Tivoli Monitoring Agent	Yes	AppServer
 IBM Tivoli Monitoring Warehouse Proxy Agent	Yes	AppServer
 IBM Tivoli Storage Manager Client Acceptor Deamon	Yes	AppServer
 IBM Tivoli Storage Manager Operations Center	Yes	J2EEServer
 IBM Tivoli Storage Manager Server	Yes	AppServer
 IBM Tivoli Storage Manager Storage Agent	Yes	AppServer
 IBM WebSphere Liberty Profile	Yes	J2EEServer
 IBM WebSphere Portal	Yes	J2EEServer
 IBM WebSphere Voice Response	Yes	AppServer
Ignore all unmatched processes	No	AppServer
InetDaemon	Yes	AppServer
JavaServer	Yes	AppServer
Legato Networker Agent	Yes	AppServer
 ManageEngine AppManager	Yes	AppServer
 ManageEngine OpManager	Yes	AppServer
 McAfee IntruShield	Yes	AppServer
 McAfee Network Security Central Manager	Yes	AppServer
 McAfee Network Security Manager	Yes	AppServer
Microsoft Biz Talk	Yes	AppServer

Table 3. Predefined custom server templates. (continued)

Name	Enabled by Default	Object Class Discovered
MySQL <b>Extension:</b> mysql.py.	Yes	DatabaseServer
Netegrity-Siteminder	Yes	AppServer
OpenView Operations Agent	Yes	AppServer
<b>Fix Pack 2</b> Oracle EM Agent	Yes	AppServer
OracleStrayProcesses	Yes	AppServer
Other Compaq Agents	Yes	AppServer
<b>Fix Pack 2</b> PaperClip Internet eXpress Package Translator System	Yes	AppServer
PostgreSQL	Yes	DatabaseServer
Print Spooler Service	No	AppServer
<b>Fix Pack 2</b> Progress OpenEdge Unified Broker	Yes	AppServer
Quadstone	Yes	AppServer
Remedy ARS	Yes	AppServer
Remote Registry Service	No	AppServer
RIM BlackBerry	Yes	AppServer
<b>Fix Pack 2</b> ServiceNow MID Server	Yes	AppServer
SiebelGateway	Yes	AppServer
SiebelServer	Yes	AppServer
SSHServer	Yes	AppServer
Symantec Anti-virus Agent	Yes	AppServer
<b>Fix Pack 2</b> Tableau Server - APIServer	Yes	AppServer
<b>Fix Pack 2</b> Tableau Server - DataServer	Yes	AppServer
<b>Fix Pack 2</b> Tableau Server - VizQLServer	Yes	AppServer
Tomcat	Yes	J2EEServer
<b>Fix Pack 2</b> Unicenter AutoSys Job Management Agent	Yes	AppServer
UNIX Builtin Services	Yes	AppServer
<b>Fix Pack 2</b> Verint Extraction Engine	Yes	AppServer
<b>Fix Pack 2</b> WebMethods Integration Server 8.2	Yes	AppServer
<b>Fix Pack 2</b> WebMethods Integration Server 9.6	Yes	AppServer

Table 3. Predefined custom server templates. (continued)		
Name	Enabled by Default	Object Class Discovered
Windows Builtin Services	Yes	AppServer
<b>Fix Pack 2</b> YSoft SafeQ CMLNode	Yes	AppServer
<b>Fix Pack 2</b> YSoft SafeQ CRSNode	Yes	AppServer
<b>Fix Pack 2</b> YSoft SafeQ ORSNode	Yes	AppServer
<b>Fix Pack 2</b> YSoft SafeQ ReplicatorNode	Yes	AppServer

## Creating and managing computer system templates

You can customize how computers systems and other devices are discovered by creating computer system templates. A computer system template can specify additional information you want to gather from a type of discovered system, and can specify the model class to use for a category of discovered device.

You can define a computer system template to specify information about a type of computer system or network device, including additional information to be gathered during discovery. Use a computer system template if you want to customize how a particular type of system or device is handled, or to gather additional information that the sensor does not discover by default.

### Adding a computer system template for an operating system

A computer system template can specify information about a particular operating system you want to discover. Use this type of computer system template if you want to discover additional detail beyond the information collected by an operating system sensor.

By creating a computer system template for an operating system, you can specify additional configuration files that you want to be included in the information gathered during discovery.

To add a computer system template for an operating system:

1. In the Functions pane of the Discovery Management Console, click **Discovery > Computer Systems**.  
The **Computer Systems** pane is displayed.
2. In the **Computer Systems** pane, click **Add**.  
The **Computer System Details** window is displayed.
3. In the **Name** field, type a name for the computer system template.  
This is the name that will appear in the list in the **Computer Systems** pane.
4. In the **Action** field, select **Discover** to specify that computer systems matching the template are to be discovered.
5. To enable the template, click **Enabled**.  
Only enabled templates are used to detect matching systems.
6. To select an icon to associate with the computer system template, click **Browse** and select the icon that you want to use.  
The icon you select will be used to represent discovered computer systems that match the template in the TADDM UI.
7. Select **Operating System** to specify that you are creating an operating system template.
8. In the **Identifying Criteria** field, select the appropriate operating system from the list.  
The template will match all computer systems running the specified operating system.
9. Click the **Config Files** tab.  
The **Config Files** page is displayed.
10. On the **Config Files** page, click **Add**.

The **Search Path for Capture File** window is displayed.

11. From the **Search Path** list, select one of the following search paths for the configuration file:

/

The root of the file system.

**\$PWD**

The current working directory of the running program.

**\$Home**

The home directory of the user ID of the running program.

**C:**

A directory on your local computer.

**%ProgramFiles%**

The program files directory.

**%SystemRoot%**

The system root directory

In the text field, type the path and file name of the configuration file you want to capture, or type an asterisk (\*) to specify all files in the selected directory.

12. To capture the contents of the specified configuration file, select **Capture file contents**. If you want to limit the amount of captured data, select **Limit size of captured file to** and specify the maximum number of bytes of the captured configuration file.
13. To search for the specified file in subdirectories of the specified location select **Recurse Directory Search** (if you use TADDM 7.3.0.3, or later), or **Recurse Directory Content** (if you use TADDM 7.3.0.2, or earlier).
14. After you have finished specifying information for the computer system template, click **OK**.
15. Optional: You can extend discovery of operating systems with commands or Jython scripts.  
For more information about extending computer system templates, see [“Extending custom server and computer system templates”](#) on page 28.

The new template is available immediately; you do not need to restart the TADDM server. The information you specified will be used to display the matching computer systems in the TADDM user interface; any configuration files you specified in the template are available in the Config Files tab for the discovered systems.

### ***Adding a computer system template for a network device***

A computer system template for a network device specifies the model object class to use for discovered SNMP devices such as routers and switches. Use this type of template if you want to provide more precise identification of a class of SNMP device, or if you want to treat an SNMP device as a computer system.

To add a computer system template for a network device:

1. In the Discovery Management Console, click **Discovery > Computer Systems**.
2. In the Computer Systems view, click **Add**.
3. In the Computer System Details window, specify the details for the new Computer System template.

The fields in this window are as follows:

**Name**

A unique name to identify the template.

**Action**

Specifies whether the template is used for discovery.

**Enabled**

Specifies whether the template is enabled. Only enabled templates are used to detect matching systems.

**Icon**

The icon used to identify the template in the Computer Systems view. Click **Browse** to see the available icons you can choose from.

## Type

Specifies the type of object matched by the template (**Operating System** or **MIB**). For a network device, select **MIB**.

## Identifying Criteria

Defines the criteria used to match discovered devices. You can define multiple criteria. To match only discovered devices that satisfy all of the defined criteria, select **All Criteria**; to match devices that satisfy any of the defined criteria, select **Any Criteria**.

Each criterion has three parts:

- The operand is the value of the discovered device to match. Select one of the following:

### Sys OID

The SNMP sysObjectID value from SNMPv2-MIB::sysObjectID (OID 1.3.6.1.2.1.1.2)

### Sys description

The SNMP sysDescr value from SNMPv2-MIB::sysDescr (OID 1.3.6.1.2.1.1.1)

- The operator specifies the kind of comparison to perform (for example, **is-greater-than** or **contains**).
- The match value specifies the value to compare to the operand.

4. Click **OK** to save the template and close the window.

The new template appears in the Computer Systems view.

**Note:** The sequence of templates in the Computer Systems view is significant. During discovery, the matching algorithm will stop when it reaches the first matching Computer System template of type MIB; if the **Action** field of this template is set to **Discover**, this template is then used for discovery. You can change the order of templates in the list by clicking the **Move Up** and **Move Down** buttons and then clicking **Save** to save your changes.

5. Optional: You can create an action class file that defines the type of ComputerSystem model object to use for the discovered device.

Typically, an action class file is used to specify the value of *type* attribute of the ComputerSystem model object for discovered devices.

**Note:** You can set any attribute value in the action class file. However, this value can be overridden. For example, the action class sets the manufacture to "IBM" but the SNMP MIB2 sensor discoveries the manufacture name is "Cisco". In this case, the sensor discovered name overrides the value provided by the action class file. If you must override a value discovered by the SNMP MIB2 sensor you must use a Jython script to do so during a discovery.

In the \$COLLATION\_HOME/etc/templates/action directory, create an XML file with the same name (not including the extension) as the Computer System template. (For example, for a template named Lucent Switch, the action class file would be named Lucent Switch.xml.)

The content of the XML file specifies the model object class and attribute values to use. The following example specifies the UnitaryComputerSystem model object class, with the type attribute set to Bridge and the manufacturer attribute set to Lucent:

```
<?xml version="1.0" encoding="UTF-8"?>
<results
  xmlns="urn:www-collation-com:1.0"
  xmlns:coll="urn:www-collation-com:1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:www-collation-com:1.0
urn:www-collation-com:1.0/results.xsd">

  <UnitaryComputerSystem array="1" xsi:type=
"coll:com.collation.platform.model.topology.sys.UnitaryComputerSystem">
    <type>Bridge</type>
    <manufacturer>Lucent</manufacturer>
  </UnitaryComputerSystem>
</results>
```

6. Optional: You can extend discovery of SNMP devices with Jython scripts.

To start a Jython script during discovery, you must first create a custom MIB Computer System template using the Computer Systems view. You can then create a descriptor file defining the Jython scripts associated with that template.

To configure a Jython script for an existing template, go to the `$COLLATION_HOME/etc/templates/commands` directory and create a new descriptor file with the same name as the template name (for example, Foundry Router). For the content of the file, specify the following:

```
SCRIPT:script_file_path
```

where `script_file_path` is the absolute path and file name of the Jython script to run. You can define multiple scripts by including multiple `SCRIPT` entries in a single descriptor file. You can include comments in the file by starting a line with the pound character (`#`).

**Note:** To use Jython scripts for discovery through an anchor, place the script in a subdirectory of the `$COLLATION_HOME/etc/templates` directory. The default location is `$COLLATION_HOME/etc/templates/extension-scripts`.

For more information about using Jython to access SNMP devices, refer to the *SDK Developer's Guide*.

### ***Editing a computer system template***

You can use the Discovery Management Console to edit a computer system template.

To edit a computer system template, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Computer Systems**.  
The **Computer Systems** pane is displayed.
2. In the **Computer Systems** pane, click **Edit**.  
The **Computer System Details** notebook is displayed. You cannot change the Name for the computer system template.

### ***Copying a computer system template***

You can create a new computer system template based on an existing one. This is done by copying a template listed in the **Computer Systems** pane and issuing it a unique name.

To copy a computer system template, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Computer Systems**.  
The **Computer Systems** pane is displayed in the workspace.
2. In the **Computer Systems** pane, select the template that you want to copy and click **Copy**.  
The **Set Name** window is displayed.
3. In the **Name** field, type the name for the new template.
4. To save the new template, click **OK**.

### ***Deleting a computer system template***

You can use Discovery Management Console to delete a computer system template.

To delete a computer system template, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Computer Systems**.  
The **Computer Systems** pane is displayed.
2. In the **Computer Systems** pane, select the template that you want to delete and click **Delete**.  
A message window is displayed.
3. To delete the template, click **Yes** in the message window.
4. To confirm the deletion, ensure that the template is not listed in the **Computer Systems** pane.

## ***Repositioning computer system template entries***

You can change the order in which computer system templates are listed in the **Computer Systems** pane. The list order is important because template matching is applied from top to bottom in the template list and stops at the first match. For example, a more generic template matches all servers of a specific type and a more specific template matches only servers that have a specific string argument.

To reposition entries in the **Computer Systems** pane, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Computer Systems**.

The **Computer Systems** pane is displayed.

2. In the **Computer Systems** pane, select the template that you want to reposition and complete one of the following steps:
  - To move the template up in the entry list, click **Move Up**.
  - To move the template down in the entry list, click **Move Down**.

## **Extending custom server and computer system templates**

Your infrastructure might contain software applications and server types, such as custom Java servers, that are not automatically categorized by TADDM. Using the Discovery Management Console, you can create custom server templates to discover and categorize these servers.

You might also want to define a custom server template for a server type that is already discovered, if you want to customize certain aspects of the way it is discovered. For example, you might want to change the display icon, or capture specific configuration files.

Defining a custom server means that you are creating a template that sets up the "membership rules" for the custom server.

During a discovery, TADDM then automatically categorizes any unknown server as a custom server of this type if the runtime information matches the criteria that you defined in the template. Any configuration files used by the custom server are also automatically captured if specified in the templates.

Creating a custom server template for an application also enables TADDM to subsequently display it as part of the topology. You can view details about the application, including the listening port, runtime information, and any config files or application descriptors that were collected.

In some cases this might not be sufficient. For example, you might also need to access the product version. By default, TADDM is unable to collect version information for arbitrary custom server applications.

You can use TADDM to extend custom server templates to collect additional information, as required, using the following approaches:

- Run commands on the target system to populate any attribute in the Common Data Model for the component

You can use this approach to set the `productVersion` attribute, for example. For more information, see the section on executing commands to populate the Common Data Model.

- Run commands on the target system and store the result as a config file for the component

One common use of this approach is to extract information from the Windows Registry. For more information about running commands to create a custom configuration file, see [“Running commands to create a custom configuration file” on page 33](#).

- Run a Jython script on the TADDM server.

You can change any information about a component. The difference between this and the first approach is where the code runs. This is more flexible than the CMD approach. The Jython based custom server extensions run on the TADDM server. These extensions can create new components and set `ExtendedAttributes`. They can also set attributes below the first level of the discovery target.



- For example, a CMD based extension can set ComputerSystem.serialNumber or any other primitive attribute of ComputerSystem. A Jython based extension can set attributes on the L2Interfaces associated with the ComputerSystem.

**Fix Pack 8** You can also configure Shell and Jython scripts together to extend a custom server template (CST). In this case, TADDM copies the user-defined Shell scripts (configured through ASDMAINSCRIPT/ASDSCRIPT tags) to the target system, run the scripts, and returns the output files to the TADDM server, and then they are parsed by a Jython script (configured via SCRIPT tag). The model objects created and return by this script file will get stored. The limitation, in this case, is that there will be no basic discovery result object (comprising of basic details like listening port, runtime information, etc.) generated or stored because of custom server templates, but which were present in other types of extensions. All data needs to be explicitly created by the Jython script to get saved. For more details, refer to the 'Using scripts to extend custom server extensions to run in script and asynchronous mode' topic in the *Using Guide*.

## Limitations

**Fix Pack 8** Extending custom server and computer system templates through Running Commands or Running Jython script is supported only in the regular discovery mode and is not supported in the script-based discovery mode. The custom server extensions extended through Shell and Jython scripts together using ASDMAINSCRIPT/SCRIPT tags, they are run in script mode or can run in asynchronous mode. For more details, refer to the 'Using scripts to extend custom server extensions to run in script and asynchronous mode' topic in the *Using Guide*.

### Migrating a script to a new Jython version

In TADDM version 7.3, two versions of Jython are available. The current Jython 2.1 is deprecated but still used by default. You can migrate to a new Jython 2.5.3.

Currently, scripts and script-based sensors use Jython 2.1 by default, although it is deprecated. To migrate to Jython 2.5.3, complete the following steps.

- To modify the script, change the following interpreters:
  - `#!/usr/bin/env ./jython_coll` to `#!/usr/bin/env ./jython_coll_253`
  - `jython_wrap` to `jython_wrap_253`
  - `cjython` to `cjython_253`
- To modify the script-based sensors, change the command  
**SCRIPT:<script.jy/py>** to **SCRIPT[com.ibm.cdb.core.jython253\_2.5.3]:<script.jy/py>**.

**Important:** Do not use different versions of Jython in one template sensor. All **SCRIPT** commands must have the same bundle, either the default one, or 2.5.3.

To configure a sensor script to use Jython 2.5.3, add the following code at its header:

```
import sys
import java
from java.lang import System
coll_home = System.getProperty("com.collation.home")
jython_home = coll_home + "/osgi/plugins/com.ibm.cdb.core."
[jython_1.0.0|jython253_2.5.3]/lib/"
sys.path.append(jython_home + "/Lib")
sys.prefix = jython_home + "/Lib"
```

- To modify the script that is run by using UniversalDataAgentConfiguration (when parserClassName is JythonParser), add the parserBundle element, like in the following example:

```
<results>
  <UniversalDataAgentConfiguration
    xsi:type="coll:com.collation.platform.model.discovery.agent.
    UniversalDataAgentConfiguration">
    ...
    <parserClassName>com.collation.platform.uda.JythonParser</parserClassName>
    <parserBundle>com.ibm.cdb.core.jython253</parserBundle>
```

```
... </UniversalDataAgentConfiguration>
</results>
```

- To modify the script interpreter that is run by using CustomTemplateSensor in CSTTemplate descriptor, add the engineId element, like in the following example:

```
<CTSTemplate>
...
<engineId>com.ibm.cdb.core.jython253</engineId>
...
</CTSTemplate>
```

## Extending custom servers

You can create a custom server template for an application to categorize the application and later display it as part of the topology. You can also view details about the application, including the listening port, runtime information, and any configuration files or application descriptors that were collected.

In some cases, however, this might not be sufficient. For example, you might also need to access the product version. By default, TADDM cannot collect version information for arbitrary custom server applications.

You can, however, extend custom server templates to collect additional information, as required, by using the following approaches:

- Run commands on the target system to populate any attribute in the IBM model for the component.

You can use this approach to set the productVersion attribute, for example.

- Run commands on the target system and store the result as a configuration file for the component.

One common use of this approach is to extract information from the Windows Registry.

- Run a Jython script on the server.

You can change any information about a component. The difference between this and the first approach is where the code runs. Additionally, you can use a Jython script to populate extended attributes and to create new model objects or relations.

To define a custom server, complete the following steps:

1. Open the Discovery Management Console.
2. Create a custom server template for the application by completing the following steps:
  - a) Click **Discovery > Custom Servers** in the sidebar.
  - b) Click **Add** to define a new custom server template.
  - c) Configure the general information and criteria for the custom server template.
  - d) Configure the custom server configuration files.

You can configure the capture of the following types of files:

- configuration files
- application descriptors
- software modules

For more information on application descriptors, see “Application descriptors” on page 66.

Software modules represent deployed application modules such as executable code or scripts that are running inside the custom server. This is an optional level of detail that you can add to the discovery of your custom server. If applicable, you can include these software modules in Business Applications for a higher level of visibility into the composition of your Business Applications.

3. Create a directive file that contains the commands to run, and add commands and scripts to the directive file, as required.

Use the format described in [Table 4 on page 31](#) when specifying commands in the directive file. [Table 5 on page 32](#) outlines the environment variables that you can use in the directive file.

Keep commands as simple as possible. If the command stops during execution, the sensor will time out and the component is not discovered.

4. Save the directive file.

The directive file must have the same name as the custom server template, and must be stored in the following directory: `$COLLATION_HOME/etc/templates/commands`. TADDMM triggers directives in this directory using the name of the custom server template.

Table 4 on page 31 describes the command format for directive files.

Table 4. Directive file format	
Directive	Description
<b>CMD:variable= path/command</b>	<p>You can define an inline command. For example:</p> <pre>CMD:productVersion=/usr/sbin/postconf  awk '/^mail_version/ {print \$3}'</pre> <p>You must always specify absolute paths to commands, and you must enclose with double quotation marks (") commands or arguments that contain spaces.</p> <p>You can use environment variables associated with the process, specified by <code>\$VARIABLE\$</code>. For example,</p> <pre>CMD:productVersion=grep versionNum \$TOMCAT_HOME\$/config/config.props   awk '{print \$2}'</pre>
<b>CMD:NOP= path/command</b>	<p>You can run the command without assigning results to a variable. For example:</p> <pre>CMD:NOP=reg export HKLM\Software\ Microsoft\InetStp c:\windows\temp\ iis.reg /y</pre>
<b>CMD:CONFCONTENT. filename= path/ command</b>	<p>You can run a command and store the results in the custom configuration file specified by filename. For example:</p> <pre>CMD:CONFCONTENT.iisREG=cmd.exe /c type c:\windows\temp\iis.reg</pre> <p>For more information, see the section on executing commands to create a custom configuration file.</p>
<b>SCRIPT: path/script</b>	<p>You can initiate Jython (.py) scripts. For example:</p> <pre>SCRIPT:path/command.py</pre> <p>When the path starts with (/) TADDMM assumes an absolute path; otherwise the path is relative from <code>\$COLLATION_HOME</code>. For more information, see the section on executing Jython scripts.</p>

Table 5 on page 32 describes the environment variables available for use in directive files.

Table 5. Directive file environment variables	
Variable	Description
<b>\$COLL_PROGPATH\$</b>	<p>This variable expands to the name of the directory where the program is located. For example, if the command line is <code>/usr/local/bin/foobar -c /etc/foobar.conf</code>, the <b>\$COLL_PROGPATH\$</b> variable expands to <code>/usr/local/bin</code>.</p> <p>You can use this variable to insulate your directive file in cases when a command is located in different directories on multiple computers.</p>
<b>\$COLL_PROGNAME\$</b>	<p>This variable expands to the fully qualified executable name. For example, if the command line is <code>/usr/local/bin/foobar -c /etc/foobar.conf</code>, the <b>\$COLL_PROGNAME\$</b> variable expands to <code>/usr/local/bin/foobar</code>. To run the appropriate command, you can use <b>\$COLL_PROGPATH\$/\$COLL_PROGNAME\$</b>.</p>
<b>\$COLL_CMDLINE\$</b>	<p>This variable expands to the entire command line, including any arguments. For example, if the command line is <code>/usr/local/bin/foobar -c /etc/foobar.conf</code>, the <b>\$COLL_CMDLINE\$</b> variable expands to <code>/usr/local/bin/foobar -c /etc/foobar.conf</code>.</p> <p>You can use this variable to find the version of the secure shell daemon (sshd) running on a system without having to know where it is installed, using the following command:</p> <pre>CMD:productVersion=\$COLL_PROGPATH\$/sshd -V 2&gt;&amp;1  awk '/version/ {print \$3}'</pre>

### Running commands to populate the Common Data Model

You can run commands on a target system to populate attributes in the Common Data Model.

For example, assume you have written a custom server template, called postfix, for the postfix mail transfer agent. You want to use the TADDM SDK to extract software information about all postfix installations so you can plan upgrades accordingly, but you notice that `productVersion` is not in the output.

You can extend the custom server template to capture this information. The `postconf` command outputs the `mail_version` string, and the third field of this line contains the version information for postfix. Therefore, you can use the following command to extract the version string:

```
$ postconf | awk '/^mail_version/ {print $3}'
```

To have TADDM run the command, create a directive file stored in `$COLLATION_HOME/etc/templates/commands/postfix` containing the following line:

```
CMD:productVersion=/usr/sbin/postconf |awk '/^mail_version/ {print $3}'
```

**Note:** Populating an attribute in the Common Data Model does not make it appear in the **Details** pane of the component, in the Discovery Management Console. The attribute is, however, stored in the database and can be retrieved using the TADDM SDK. To have the attribute appear in the user interface, use a configuration file extension, as described in the section on running commands to create a custom configuration file.

- An attribute in the Common Data Model, populated by a Custom Server Extension, is displayed in the Discovery Management Console only if the Discovery Management Console has been designed to display the given attribute.
- Populating attributes does not change which attributes can be displayed by the Discovery Management Console.
- Output that is captured by the CMD : CONFCCONTENT directives is displayed on the **Configuration Files** tab in the Discovery Management Console if the target type that is discovered has this tab in the Discovery Management Console.

### ***Running commands to create a custom configuration file***

You can run commands on a target system and store the results in a custom configuration file.

You might want to do this to record a portion of the Windows Registry in a configuration file, for instance. You can save the results in a custom configuration file to access the information using the Discovery Management Console.

**Note:** Output that is captured by the CMD : CONFCCONTENT directives is displayed on the **Configuration Files** tab in the Discovery Management Console if the target type that is discovered has this tab in the Discovery Management Console.

For example, assume that you wrote a custom server template called HomePageWebServer that subcategorizes Microsoft IIS. You can create a directive file with the same name, stored in the \$COLLATION\_HOME/etc/templates/commands directory, containing the following line:

```
CMD:NOP=reg export HKLM\Software\Microsoft\InetStp c:\windows\temp\iis.reg /y
CMD:CONFCCONTENT.iisREG=cmd.exe /c type c:\windows\temp\iis.reg
```

When the commands in the directive file are run, the results of the Registry export (reg export) are stored in a configuration file with the name iisREG.


**Limitation:** If you use the CMD : CONFCCONTENT directive to extend the existing sensor, custom application server sensor is run in addition to the existing sensor. In such case, the configuration files are stored only for the sensor that stores the results last. For example, if you use the directive to extend the Oracle sensor, both the Oracle and custom application server sensors are run. If the Oracle sensor stores the results as last, the configuration files are collected only for this sensor. If you want to store the configuration files that are captured by both sensors, you can use prioritization rules to specify which sensor to store as last. For example, to store the configuration files that are defined for both the Oracle sensor and the extension that you created, you must first edit the extension to also collect all files that the Oracle sensor stores. Then, you must set the custom application server sensor to have priority over the Oracle sensor. As a result, the configuration files that are stored for the Oracle sensor and extension are captured by the custom application server sensor. As an alternative, you can use custom template sensor extension. The only difference in comparison to the custom application server sensor is that the results of the existing sensor, for example the Oracle sensor, are included in the custom template sensor extension. Therefore, you do not need to edit the extension, but the prioritization rules still apply.

For more information about the prioritization rules, see [“Adding prioritization rules to your configuration items \(model objects\)” on page 77](#).

### ***Running Jython scripts***

You can extend custom server and computer system templates by calling Jython scripts (.py extension) on a target system.

TADDM automatically detects the script language and passes the context of the custom server or computer system to the scripting language using a hashmap. This enables Jython to manipulate Java objects.

The detailed structure  along with the sample script and its explanation is described in the *Custom server extensions API* section of the *TADDM SDK Developer's Guide*. However, initializing the sensor helper section is different for extensions of the custom server template and computer system template. The following sections provide description of these differences.

### sensorhelper section of the Jython script

The Jython script that you use to extend the custom server and computer system templates contains the sensorhelper section. This section initializes the sensor tools Python module with information about the target of the discovery. Depending on which template you use to extend the discovery scope, it has the following form:

- Initializing sensorhelper for a custom server:

```
(os_handle, result, appserver, seed, log, env)=sensorhelper.init(targets)
```

- Initializing sensorhelper for a computer system:

```
(os_handle, result, computersystem, seed, log)=sensorhelper.init(targets)
```

**Note:** Do not change the order of the elements in this section of the script. Otherwise, it fails.

### Target map objects for the custom server

The following table lists and describes the objects that are available in the targets map for the custom server.

Table 6. Target map objects for the custom server	
Object	Description
os_handle	<p>This object is an implementation of the TADDM Os abstraction layer. You can use it to run remote commands.</p> <p>The following method is available:</p> <ul style="list-style-type: none"><li>• executeCommand(String cmd): Runs a command on the remote target and returns the output as a string.</li></ul>
result	<p>The CustomAppServerResult object</p> <p>The following methods are available:</p> <ul style="list-style-type: none"><li>• getServer(): Returns the AppServer that is being discovered.</li><li>• setServer(AppServer app): Sets the AppServer in the resulting object for persistence.</li><li>• addExtendedResult(ModelObject mo): Adds a CDM ModelObject (or subclass) to the result object so that it can be persisted in the data store. This ModelObject does not need to be related to the target for which the sensor was started.</li><li>• addConfig(AppConfig apconfig): Adds a configuration file to the result object for persistence.</li></ul>
appserver	<p>The AppServer that is being discovered</p> <p>The available methods are listed in the TADDM Data Dictionary. For details, see the <i>TADDM Data Dictionary</i> topic in the <i>TADDM SDK Developer's Guide</i>.</p>

<i>Table 6. Target map objects for the custom server (continued)</i>	
<b>Object</b>	<b>Description</b>
seed	<p>The CustomAppServerSeed object</p> <p>The following methods are available:</p> <ul style="list-style-type: none"> <li>• getSessionIp(): Returns the IP address that TADDM used to connect to the computer system where the AppServer that is being discovered is running.</li> <li>• getPrimarySapIpAddr(): Returns the IP address to which the discovery target is bound. If it is bound to all interfaces, then the session IP is returned.</li> </ul>
log	<p>The object used for writing to sensor logs.</p> <p>The following methods are available for various log levels, starting with most severe:</p> <ul style="list-style-type: none"> <li>• fatal</li> <li>• error</li> <li>• warning</li> <li>• info</li> <li>• debug</li> <li>• trace</li> </ul>
env	<p>The Java HashMap environment object. The keys are the processes environment variables. The values are those of the variables.</p>

### Target map objects for the computer system

The following table lists and describes the objects that are available in the targets map for the computer system.

<i>Table 7. Target map objects for the computer system</i>	
<b>Object</b>	<b>Description</b>
os_handle	<p>This object is an implementation of the TADDM Os abstraction layer. You can use it to run remote commands.</p> <p>The following method is available:</p> <ul style="list-style-type: none"> <li>• executeCommand(String cmd): Runs a command on the remote target and returns the output as a string.</li> </ul>

Table 7. Target map objects for the computer system (continued)	
Object	Description
result	<p>The ComputerSystemResult object</p> <p>The following methods are available:</p> <ul style="list-style-type: none"> <li>• <code>getComputerSystem()</code>: Returns the ComputerSystem being discovered.</li> <li>• <code>setComputerSystem()</code>: Sets the ComputerSystem in the result object for persistence.</li> <li>• <code>addExtendedResult(ModelObject mo)</code>: Adds a CDM ModelObject (or subclass) to the result object so that it can be persisted in the data store. This ModelObject does not need to be related to the target for which the sensor was started.</li> </ul>
computersystem	<p>The ComputerSystem that is being discovered</p> <p>The available methods are listed in the TADDM Data Dictionary. For details, see the <i>TADDM Data Dictionary</i> topic in the <i>TADDM SDK Developer's Guide</i>.</p>
seed	<p>The ComputerSystemSeed object</p> <p>The following method is available:</p> <ul style="list-style-type: none"> <li>• <code>getIpAddress()</code>: Returns the IP address that TADDM uses to discover the target.</li> </ul>
log	<p>The object used for writing to sensor logs.</p> <p>The following methods are available for various log levels, starting with most severe:</p> <ul style="list-style-type: none"> <li>• fatal</li> <li>• error</li> <li>• warning</li> <li>• info</li> <li>• debug</li> <li>• trace</li> </ul>

### Running the script

To run the script, you must create the directive file, and include the name of the Jython script, for example:

```
SCRIPT:mymyscript.py
```

For more information about the format of the directive file, see [“Extending custom servers” on page 30](#).

For example, you can run the Jython script `mymyscript.py` by including the following command in the directive file:

```
SCRIPT:mymyscript.py
```



## Using scripts to extend custom server extensions to run in script and asynchronous mode

You can use shell scripts or Jython scripts to extend a custom server template (CST).

**Fix Pack 8** When these are used, then script-based discovery is performed where TADDM copies the user-defined scripts (ASDMAINSCRIPT/ASDSCRIPT) to the target system, run the scripts, and returns the output files to the TADDM server. The output generated by these script files needs to be parsed by a Jython script file, that must be created by the user which will be configured via the SCRIPT tag in the directive file. Only the model objects if any, created and returned by this script file will get stored.

In the case of Asynchronous discovery mode (ASD), the user can generate a request package and these scripts configured with **ASDMainScript** and **ASDScript** tags, will get bundled in it. Users can then run the package at target and get a result file generated like the ASD process for other sensors. This result file will be copied to the asdd directory on the TADDM server by the user, and then when ASD discovery is invoked via the Discovery Management Console, it will process these files. The Jython script (tagged with SCRIPT tag) mentioned in the directive file will process the output result and will generate result objects as per user coding in that Jython script. Hence, in this mode of extension define one main script (ASDMAINSCRIPT) for a given custom server template. If required, more than one additional scripts (ASDSCRIPT) that are used by the main script can be defined as well. Along with these, define another Jython script (SCRIPT), that will parse the outputs of the scripts run on targets and create model objects and return them for storage.

**Important:** **Fix Pack 8** Extending custom server extensions by using Shell and Jython script make it run as a script-based sensor. This extension is also available in asynchronous discovery mode. However, in this mode of extension, only the model objects created and returned by the Jython script file will get stored. This implies there is **no** basic discovery result object comprising of listening port, runtime information, etc, will get generated because of custom server templates which is present in other types of extensions..

To start a custom server template using a script, define one main script for a given custom server template. If required, you can define one or more additional scripts which are used by the main script.

### Definition of CST extended with scripts

A CST can have a command file defined in \$COLLATION\_HOME/etc/templates/commands directory. The command file name has the same name as the CST name. Add the following commands to the command file to extend the CST with scripts:

#### ASDMAINSCRIPT

This command specifies the main script file name. It specifies the script to be copied and started on the target system.

#### ASDSCRIPT

This command specifies additional script file names if required. It specifies scripts to be copied and used by the main script on the target system.

**Fix Pack 8** **SCRIPT**

This command specifies the Jython script file name. It specifies the script that parses the outputs of scripts run on targets and creates model objects as well. Only the model objects, if any, created and returned by the Jython script file will get stored.

For Asynchronous discovery, the directive filename or the custom server template name must not have space in its name.

### Command syntax

Add the following commands to the command file:

#### ASDMAINSCRIPT:os\_discriminator:script\_relative\_path

The variable *os\_discriminator* defines the operating system where you can run the script. The following values are valid:

- AIX
- LINUX
- SOLARIS
- UNIX
- WINDOWS
- ALL: Use this value to indicate all operating systems.

The variable *script\_relative\_path* defines a relative path to a script starting from the \$COLLATION\_HOME directory. Place the scripts in a subdirectory of \$COLLATION\_HOME/etc/templates/commands directory.

### **ASDSCRIPT:os\_discriminator:script\_relative\_path**

The same definitions as described previously also apply to the ASDSCRIPT command.

### **Example of a command file**

A script set is a set of scripts defined with the same *os\_discriminator* attribute. Each script set must have one main script (ASDMAINSRIPT) and can have if required one or more additional scripts (ASDSCRIPT). The CST extended by scripts chooses the most specific script set for the discovered operating system.

The following example shows a command file for a CST extended with scripts:

```
ASDMAINSRIPT:AIX:etc/templates/commands/scripts/scriptAix.sh
ASDSCRIPT:AIX:etc/templates/commands/scripts/myTest3.sh
ASDMAINSRIPT:UNIX:etc/templates/commands/scripts/scriptUnix.sh
ASDSCRIPT:UNIX:etc/templates/commands/scripts/myTest.sh
ASDSCRIPT:UNIX:etc/templates/commands/scripts/myTest2.sh
SCRIPT: etc/templates/commands/scripts/myOutputParser.py
```

This command file defines two script sets: one for AIX only, and one for UNIX and similar operating systems.

**Fix Pack 8** In the ASDMAINSRIPT file, to invoke the other script for example, myTest . sh, follow the pattern as used in the normal script sensors, for example, echo SCRIPT:myTest . sh, only the file name must be mentioned without any directory path.

## **Script generated files**

TADDM copies the defined scripts to the target system and runs the main script. The main script is started in the root system directory, but it is located in a temporary directory. To get the actual script location, use the *dirname \$0* command.

The scripts and files generated by the script, which are stored in the temporary location on the target system, are returned to the TADDM server. The files are placed in the \$COLLATION\_HOME/var/asdd/{runId}/{targetIp}/{sensorOsgiId} directory. The *sensorOsgiId* name consists of the custom application server sensor (CustomAppSever) identifier and the template name. For example, com.ibm.cdb.discover.sensor.app.customappserver\_7.1.0.JavaServer for a JavaServer CST.

### **Add a Jython extension to a CST extended with scripts**

To parse data gathered by a CST extended with scripts, you can call Jython scripts (.py extension) automatically during a discovery. To add a Jython extension to a script file, you must add the following command to the directive file:

#### **SCRIPT:jythonscript\_relative\_path**

The directive file must have the same name as the CST and must be stored in the following directory: \$COLLATION\_HOME/etc/templates/commands

TADDM passes the context of the custom application server sensor to the script language using a hashmap or script target map. This method enables Jython to manipulate Java objects. The script target

map has predefined objects that can be used by the script for processing and for passing back results. The following objects are available in the script target map for a CST extended with scripts:

- outputs - the List <OutputDataSet> object
- systeminfo - the system information object
- seed - the CustomAppServerSeed object
- result - the CustomAppServerResult object
- environment - the Java HashMap environment object containing the Map <String, String> object

The following example shows a sample of Jython source code:

```
...
log = LoggerFactory.getLogger("com.ibm.cdb.discover.sensor.CustomAppServerScriptSensor")

result = scripttargets.get("result")
seed = scripttargets.get("seed");
env = scripttargets.get("environment")
systeminfo = scripttargets.get("systeminfo")
outputDataSetList = scripttargets.get("outputs")

# just print the content of output data set list
for outputDataSet in outputDataSetList:
    if not outputDataSet.isValid():
        log.error("Not valid output data set")
        continue
    for outputData in outputDataSet.iterator():
        if not outputData.isValid():
            log.error("Not valid output data")
        try:
            log.info(outputData.getValue())
        except ExecutionException, e:
            log.error("ExecutionException", e)
...

```

**Fix Pack 8** For information on how to create model objects using extensions, refer to the 'Creating new objects and relationships through custom server extensions' topic in the *Using Guide*.

**Tip:** For more examples of using the Jython script to extend a discovery scope, see the *Extending sensor discovery scope with Simplified Model* topic in the *TADDM SKD Developer's Guide*.

### ***Creating new objects and relationships through custom server extensions***

You can create new objects and relationships through custom server extensions.

To create new objects and relationships, call the `addExtendedResult()` method on the result object handed to the script. The `addExtendedResult()` method call takes a `ModelObject` as the parameter. It can be called repeatedly if you want to create more than one new object. At least one of the naming rules, defined by the Common Data Model, must be set on any new `ModelObjects` that are added to the `ExtendedResult` for storage. Failure to do so causes the sensor to fail with a "Storage Error" because the TADDM reconciliation engine was unable to generate a Globally Unique Identifier (GUID) for the given `ModelObject`. For more information, see the Common Data Model documentation, located in the `dist/sdk/doc/model/CDMWebsite.zip` file of your TADDM installation DVD.

Custom server extensions are started when the `CustomComputerSystemSensor` or `CustomAppServerSensor` runs. If the sensor does not run, the custom server extension does not start, and the new objects and relationships are not created.

**Fix Pack 8 CTS vs CST vs CSX**

This following table describe the difference between Custom Template Sensor (CTS), Custom Server Template (CST), and Custom Server Extension (CSX).

**CTS vs CST vs CSX**

<i>Table 8. Difference between Custom Template Sensor (CTS), Custom Server Template (CST), and Custom Server Extension (CSX)</i>			
	<b>Custom Template Sensor</b>	<b>Custom Server Template</b>	<b>Custom Server Extension</b>
Definition	The custom template sensor is used to analyze and enhance the information collected by any of the existing TADDM sensors.	The custom server template is used to discover and categorize servers that are not by default, supported by TADDM.	Although details like listening port, runtime information, configuration files, application descriptors, etc., can be collected using Custom Server Templates (CST) but when additional information is needed such as product version, etc., extension scripts (CSX) are used.
Sensor Run	CustomTemplateSensor	CustomAppServerSensor	CustomAppServerSensor
Use Cases	To get extra information, which is not currently obtained from an existing TADDM sensor.	<ul style="list-style-type: none"><li>• Discover and categorize servers that are not by default, supported by TADDM.</li><li>• Capture configuration files specified in the Custom Server template.</li><li>• Creating a custom server template for an application also enables TADDM to subsequently display it as part of the topology. You can view details about the application, including the listening port, runtime information, and any config files or application descriptors that were collected.</li></ul>	<ul style="list-style-type: none"><li>• Defining CSX on CST allows to obtain additional customized information, such as product version, etc., besides what is collected by CST.</li><li>• To create new objects and relationships through the use of extension through the Jython script.</li></ul>

*Table 8. Difference between Custom Template Sensor (CTS), Custom Server Template (CST), and Custom Server Extension (CSX) (continued)*

	<b>Custom Template Sensor</b>	<b>Custom Server Template</b>	<b>Custom Server Extension</b>
Ways to create	To create a Custom Template Sensor, three configuration files are needed: template.xml, matcher.py, and sensor.py.	Creating Custom Server Templates involves defining templates on the Discovery Management Console.	<p>There are three possible ways to create the extensions:</p> <ol style="list-style-type: none"> <li>1. Run commands on the target system to populate any attribute in the IBM model for the component.</li> <li>2. Run commands on the target system and store the result as a configuration file for the component.</li> <li>3. Run Jython script on the server and change or add any information about a component.</li> </ol> <p>An additional way is using shell and Jython script using ASDMAINS script and SCRIPT tags where TADDMM copies the defined scripts to the target system, runs the scripts and returns the output files to the TADDMM server which are then processed by a Jython script. However, it comes with the limitation of having to create all model objects and data to be saved explicitly. This is run in script or asynchronous discovery mode.</p>
Limitations	Cannot be run in script-mode.	Cannot be run in script-mode.	Cannot run in script-mode except using Shell and Jython scripts using ASDMAINS script and SCRIPT tags which runs in the script or asynchronous discovery mode but comes with limitations.

## **Anchors and gateways**

You can use anchors and gateways to extend discoveries to restricted network zones and to offload some of the discovery process from the TADDMM server to improve the overall discovery performance.

The TADDMM server uses SSH protocol to directly communicate with computer hosts and other components that it discovers. However, there are two cases when the server must communicate through a proxy to collect system information:

- When using a firewall between the TADDM server and other sections of your network.
- When discovering and collecting information from Windows systems.

### Requirements information

- For information on the requirements for Windows gateways, see the *Windows gateways* topic in the *TADDM Installation Guide*.
- If you use anchors on the Windows operating system, for requirements see the *Windows gateways* topic in the *TADDM Installation Guide*.

**Restriction:** Anchors are supported on Cygwin 64-bit edition on Windows Server 2012 x64 and Windows Server 2008 x64. However, the discovery user and the user that starts the service must be the same.

- If you use anchors on the Linux or AIX operating systems, you can use only the systems that are supported by the TADDM server. For hardware requirements, see the *Discovery server* topic in the *TADDM Installation Guide*.

### Running a discovery that requires anchors

In the following example, the discovery scope is the set of scope elements selected for discovery in the **Run Discovery** window. See [“Running a basic discovery” on page 11](#) for more information.

When running a discovery that requires anchors, ensure that each anchor is included in the discovery scope. For example, to discover a target that is in a scope set (for example, `scopeset1`) assigned to an anchor, both the anchor and the scope set (`scopeset1`) must be included in the discovery run. The scope set or sets that are assigned to each anchor should only include the IP addresses accessible by the anchor. In addition, if the scope sets assigned to the root server are restricted, they should include only the set of IP addresses that the TADDM server can access directly, including the IP addresses of any defined anchors.

### Balancing the load during a discovery

Use the following properties to balance the load during a discovery:

#### **com.ibm.cdb.discover.agents.max**

defines the maximum number of agents to be run simultaneously on an anchor. This property can be scoped to a profile or a specific anchor server.

The default value is the value of the `com.collation.discover.dwcount` property.

#### **com.ibm.cdb.discover.workitem.cooldown**

is a general property and defines the amount of time, specified in seconds, that a work item waits before it is processed again.

The default value is 30.

### Configuring anchors

You can configure anchors for discovery when a firewall is present.

IP devices must respond to pings from either the TADDM server or an anchor in order to be discovered. Because most firewalls are not configured to forward pings, the TADDM server is unable to ping systems behind a firewall and is unable to discover them. To enable discovery through a firewall, an anchor must be identified to assist the TADDM server with the discovery process.

The anchor must be located in the same network section as the target for discovery and meet the same software requirements as the TADDM server.

Before you can discover systems that have a firewall between them and the TADDM server, the TADDM server must allow SSH traffic to the anchor. Make sure that your network administrator configures the firewall to enable SSH traffic between the TADDM server and the anchor. You must use SSH version 2 network protocol when exchanging the data.

On Linux and UNIX systems, the discovery service account must have root execution permission for the **nmap** command. Make sure that the following line exists in the `/etc/sudoers` configuration file:

```
TADDM_userid ALL=(ALL) NOPASSWD:nmap_path
```

where

- *TADDM\_userid* is the TADDM discovery service account on the anchor system.
- *nmap\_path* is the full path to the location of the **nmap** command.

If the `sudoers` file contains a `Defaults requiretty` line, comment it out.

The anchor is created by the AnchorSensor through an SSH session that connects to the system defined as the anchor. The user for the SSH session is the first Computer System (or Computer System (Windows)) entry in the Access List that completes a successful connection. On the anchor system, the home directory for this user must be writable by the user and have at least 1.2 GB of free space. The TADDM files, including the Java SDK, are transferred to this directory by using **scp** and extracted. Since these files contain executable code, you must either disable antivirus programs or configure them to allow the anchor user to transfer and extract this code. Anchors are also automatically redeployed by the AnchorSensor after TADDM maintenance changes, such as fix packs.

If the network connection between the TADDM server and the anchor system is slow, or if the TADDM server and the anchor server are far apart, the AnchorSensor might time out before it completes the creation of the anchor. The default timeout value is 20 minutes. To change the timeout value for the AnchorSensor to another value, modify the setting for `com.collation.discover.agent.AnchorSensor.timeout` value in the `COLLATION_HOME/etc/collation.properties` file. The value is in milliseconds, so the default value is 1200000, which equals 20 minutes.

After the firewall setup is complete, define the anchor by using the Discovery Management Console. When you define the anchor, you must include it in the scope of the root server. The scope of the anchor must be restricted to the systems in that network section. When discovery is initiated from the Discovery Management Console, the TADDM server deploys the necessary files to the anchor. After the files are deployed, the anchor runs the discovery and returns the results to the TADDM server.

If there are multiple zones or firewalls, you must specify at least one anchor in each adjacent zone so that communications can be relayed from each anchor across each firewall. To do this, SSH traffic must be enabled between each pair of adjacent anchors, starting with the root server. Each anchor in the next adjacent network subnet must be included in the scope of the anchor in the previous subnet. Anchors chained in this way must be running on the same operating system type.

**Note:** If more than one anchor is specified in a single scope, and there is connectivity between them, TADDM attempts to start anchor chains on these anchors. This behavior may result in various error messages on the GUI and in the logs, even if the anchor is deployed correctly.

See [“Adding an anchor or gateway” on page 46](#) for information about defining anchors using the Discovery Management Console.

Also note that the TADDM user interface does not indicate which NAT zone an object is in. To avoid confusion, make sure hosts with the same IP address in different NAT zones have different host names, which makes it possible to distinguish them. Assign different domains (for example, `nat1.lab.company.com`, `nat2.lab.company.com`) to each NAT zone. This ensures that the fully-qualified host names from different NAT zones are unique. Note that if the same DNS server is used for different NAT zones with identical subnet addresses, then different DNS views must be used for each zone.

**Note:** When an anchor or other dual-homed host is discovered through an L1 discovery, the host might be displayed as a duplicated entry in the physical infrastructure tree. This duplication occurs because the host is discovered twice, once by TADDM and once by the anchor, and the L1 discovery does not provide enough information to reconcile the two discoveries. To reconcile the two entries as one host, run an L2 or L3 discovery.

If more than one domain server, in synchronization server deployment, or discovery server, in streaming server deployment, use the same machine as an anchor on which to perform simultaneous discoveries, the following changes must be made:

- Set the anchor port.
  1. In the **Anchors and Gateways** pane of the Discovery Management Console, click **Set Anchor Port**. The **Edit Port Number** window is displayed.
  2. In the **Port No.** field, type the port number. Ensure that the port number is different for each TADDM server.
  3. Click **OK**.
- Set the anchor directory.
  1. Open the \$COLLATION\_HOME/etc/collation.properties file.
  2. Set the **com.ibm.cdb.taddm.anchor.root** property value to the anchor directory name. Ensure that the property is not commented out and that the directory is different for each TADDM server.

TADDM sets a location tag attribute for each configuration item (CI) that is created on the TADDM server. To set the location tag attribute for CIs that are created on an anchor, configure the `anchor_location_n` attribute in the \$COLLATION\_HOME/etc/anchor.properties file. The following sample entries from the `anchor.properties` file indicate how location information for anchors is set:

```
anchor_host_1=192.168.1.13
anchor_scope_1=FIRST_SCOPE
anchor_zone_1=FIRST_ZONE
anchor_location_1=FIRST_LOCATION
anchor_host_2=192.168.2.22
anchor_scope_2=SECOND_SCOPE
anchor_location_2=SECOND_LOCATION
Port=8497
```

If a location tag is not specified for an anchor, the location of each of the CIs that are created on the anchor is set to the location that is specified for the TADDM server to which the CIs are connected. If the location tag is not specified for the anchor or the TADDM server, no location information is set for that CI.

### **Configuring for discovery through NAT firewalls**

You can create NAT zones to support discovery of hosts on a private network behind a NAT firewall.

TADDM supports the discovery of hosts on a private network that uses network address translation (NAT). A private network, as defined by RFC 1918, uses private IP addresses that fall within one of the following address blocks:

- 10.0.0.0/8 (10.0.0.0 through 10.255.255.255)
- 172.16.0.0/20 (172.16.0.0 through 172.31.255.255)
- 192.168.0.0/16 (192.168.0.0 through 192.168.255.255)

Because different private networks accessible through NAT firewalls might use the same addresses, discovery of objects using these addresses might not be able to identify hosts on different private networks as distinct. For example, two different objects might each be assigned the private IP address 10.10.10.3 because they exist on different private networks. When discovered through a NAT firewall, one object would overwrite the other in the TADDM database.

You can avoid this problem by creating *NAT zones*, which are arbitrary strings you define to identify each private network reachable through NAT. When TADDM discovers an IP address within a private network associated with a NAT zone, it includes the zone string as part of the stored IP address object. This ensures that objects from different private networks are identified unambiguously and stored separately, even if they have the same IP addresses on their respective networks.

To configure for discovery through a NAT firewall, complete the following steps:

1. Create an anchor using the Discovery Management Console.  
To support NAT discovery, you must specify a scope.



2. Edit the `$COLLATION_HOME/etc/anchor.properties` file with a text editor.
3. Find the `anchor_host_n` and `anchor_scope_n` entries (where *n* is a number) corresponding to the newly added anchor.
4. Add a corresponding `anchor_zone_n` entry.

For example, if you created an anchor host with the address 9.43.73.184 and the scope QA\_SCOPE, the modified entry might be as follows:

```
#Last modified on:
#Mon Feb 16 16:19:52 PST 2009
anchor_host_1=9.43.73.184
anchor_scope_1=QA_SCOPE
anchor_zone_1=QA_ZONE
port=8497
```

**Note:** Changing a NAT zone name can cause creation of duplicate objects associated with the new name. Therefore, when naming a NAT zone, choose a string that is a symbolic and meaningful description of the private network, rather than a name derived from the IP address or hostname of the anchor (which might change). After you have defined the NAT zone and used it for discovery, avoid changing it.

5. Run a discovery on the NAT scope and a scope containing the anchor.

In the previous example, this would be a discovery on the QA\_SCOPE scope and a scope containing the 9.43.73.184 anchor address.

## Configuring Windows gateways

You can configure Windows gateways for discovery when a firewall is present.

Both an anchor and a gateway are needed to discover Windows systems behind a firewall. The TADDM server uses SSH to communicate with the anchor behind the firewall. The anchor in turn uses SSH to communicate with the Windows gateway. The Windows gateway then uses Windows Management Instrumentation (WMI) to discover the Windows targets. There must be one Windows gateway located in each network zone in which you want to discover Windows systems.

See [“Adding an anchor or gateway” on page 46](#) for information about defining Windows gateways using the Discovery Management Console.

If you are using both an anchor and a gateway on the same system, to resolve Cygwin issues add the following entry to the `Collation.properties` file:

### **com.collation.platform.session.GatewayForceSsh**

Specifies whether to force the gateway to act independently of the anchor. Valid values are *true* and *false*. Set the value to *true*. When the value is set to *true*, an SSH session is used to transfer traffic between the gateway and anchor rather than a local session.

## Configuring for discovery through a firewall without an anchor

In environments that have many firewalls, deploying an anchor to each firewall zone might not be practical.

In these situations, it might be easier to open up ports in the firewall rather than deploying anchors. For non Windows platforms, the list of ports that must be opened are those which are needed for application discovery. Because application ports can be configured, there is not a predefined list of ports. Typically, opening all ports above 1024 should be sufficient.

For Windows systems, in addition to the application ports, the file sharing ports and restricted RPC ports must be opened:

- 139 RCP Enable NetBIOS Session Service
- 445 TCP Enable SMB over TCP
- 137 UDP Enable NetBIOS Name Service
- 138 UDP Enable NetBIOS Datagram Service
- 135 TCP Enable DCOM

- 5000 TCP Enable RPC
- 5001 TCP Enable RPC
- 5002 TCP Enable RPC
- 5003 TCP Enable RPC
- 5004 TCP Enable RPC
- 5005 TCP Enable RPC
- 5006 TCP Enable RPC
- 5007 TCP Enable RPC
- 5008 TCP Enable RPC
- [...]
- 5099 TCP Enable RPC
- 5100 TCP Enable RPC

To limit the range of ports that Microsoft uses for RPC, see <http://support.microsoft.com/kb/154596> for more information.

**Fix Pack 2** In TADDM 7.3.0.2, and later, you can use PowerShell session to discover Windows target systems. For this session, port 5985, or 5986 must be opened. The ports listed previously are not required. If your firewall is configured to allow only limited communication, for example it allows only the PowerShell ports, you need to configure the Ping sensor for the discovery to be successful. Add the `com.collation.pingagent.ports` property to the `collation.properties` file, and set the value to 5985, or 5986, or both. For details, see the *Configuring the collation.properties file entries* topic in the *Ping sensor* section of the *TADDM Sensor Reference*.

### ***Adding an anchor or gateway***

You can use the Discovery Management Console to add an anchor or a gateway.

To add an anchor or Windows gateway, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Anchors and Gateways**.  
The **Anchors and Gateways** pane is displayed.
2. In the **Anchors and Gateways** pane, click **Add**.  
The **Add Anchor** window is displayed.
3. From the **Type** list, complete one of the following steps:
  - To add an anchor, select **Anchor**.
  - To add a Windows gateway, select **Windows Gateway**.
4. To identify the anchor or Windows gateway, complete one of the following steps:
  - To set by the IP address, click **Address** and then type the IP address in the **Address** field.
  - To set by the host name, click **Host Name** and then type the host name in the **Host Name** field.
5. In the Scope to search for host section, complete one of the following steps:
  - To include the entire discovery scope, click **Entire scope**.
  - To restrict the scope used by the anchor, click **Limit to selected scope sets** or **Limit to selected scope groups** and then select the scope sets or scope groups that you want to include.
6. To save the anchor or Windows gateway, click **OK**.

After you add an anchor, you must include the IP address or hostname for it in the discovery scope. See [“Configuring a scope” on page 2](#) for more information.

### ***Editing an anchor or gateway***

After you add an anchor or Windows gateway, you cannot change its type, IP address, or host name, but you can edit the scope.

To edit the scope of an anchor or Windows gateway, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Anchors and Gateways**.  
The **Anchors and Gateways** pane is displayed.
2. In the **Anchors and Gateways** pane, select the anchor or gateway that you want to edit, and click **Edit Scope**.  
The **Add Anchor** window is displayed.
3. To change the scope settings, complete one of the following steps:
  - To use the access information across all components of the defined scope set, click **Entire scope**.
  - To restrict the application of specific access information to certain systems, click **Limit to selected scope sets** or **Limit to selected scope groups** and then select the scope set or scope group that you want to restrict access to.
4. To save your changes, click **OK**.

### ***Deleting an anchor or gateway***

You can use the Discovery Management Console to delete an anchor or Windows gateway.

To delete an anchor or Windows gateway, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Anchors and Gateways**.  
The **Anchors and Gateways** pane is displayed.
2. In the **Anchors and Gateways** pane, select the anchor or Windows gateway that you want to delete and click **Delete**.  
A message window is displayed.
3. To delete the anchor or Windows gateway, click **Yes** in the message window.
4. To confirm the deletion, ensure that the anchor or Windows gateway is not listed in the **Anchors and Gateways** pane.

### ***Setting an anchor port***

If the default anchor port is in use by another application, you must set a new port number for the anchor.

**Important:** The anchor port is a global setting for all anchors. Therefore, when you set a new port number for an anchor, the new port number is assigned to all anchors.

To set the anchor port, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Anchors and Gateways**.
2. In the **Anchors and Gateways** pane, select an anchor and click **Set Anchor Port**.
3. In the **Port No:** field, type the port number for the anchor.
4. To save the port number, click **OK**.

### ***Stopping the anchor***

A discovery might include one or more anchors. The anchors should stop running when they complete, or if they time out. If you want to shut down the TADDM server, you must also shut down each anchor. If you do not shut down each anchor, it can lead to unexpected behavior, including bad performance of certain discoveries.

If the anchor does not stop on its own, complete the following steps from the Discovery Management Console:

1. Verify that the anchor processes are no longer running by typing the following command:

```
% ps -ef |grep -i anchor
```

This command identifies any anchor processes that are still running. You should get the following similar output:

```
coll 23751 0.0 0.0 6136 428 ? S Jun02 0:00 /bin/sh
local-anchor.sh 8494 <more information here>
```

2. Stop the process by typing the following command:

```
- % kill -9 23751
```

You should get either no output or something similar to the following:

```
root 13561 13486 0 16:19 pts/0 00:00:00grep -i anchor
```

## Discovery schedules

You can schedule discoveries to ensure that information presented in the Discovery Management Console is always current and accurate.

In most cases, partition your environment into operational groups and perform discoveries on these subsets of your organization. This reduces the time it takes to complete a particular discovery, and takes into account that different sections of your environment change at different rates.

If you create a schedule to run a discovery, it binds the current scope to that schedule. Later, if you want to add a new entry to the scope, you must delete the schedule and create a new one. You can schedule a discovery to perform the following tasks:

- Identify operational groups within your environment. Different sections of your environment are likely to have different rates of change. By identifying operational groups on your network by IP addresses, IP address ranges, and subnets, you can schedule partitions of your infrastructure to have different discovery schedules.
- Check the discovery history to determine how long it typically takes to complete different types of discoveries in your environment.

Discovery schedules cannot overlap. The first discovery must complete before a new discovery can begin. If a discovery is scheduled to start before an existing discovery finishes, the new discovery does not start and an error is logged.

Check the discovery history to estimate the typical completion time for different discoveries, so that you can prevent potential schedule overlaps.

- Schedule discoveries based on the operational groups that you identified. Configure most of your scheduled discoveries to refresh a subset of your topology. For example, depending on the size of your environment and your operational needs, you can schedule a full discovery once every 24 hours, or complete a discovery of the application tier once every six hours.

When you create a discovery schedule, you specify the start time and the frequency of discoveries. You can also define the scope of a particular discovery by selecting the scope elements (subnets, IP addresses, or ranges), components, or views to include in the discovery.

### Related reference

[“Schedule pane” on page 126](#)

You can view schedule information in the **Schedule** pane.

## Adding a discovery schedule

Adding a schedule instructs the server to run a discovery at the defined time.

To add a discovery schedule, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Schedule**.

The **Schedule** pane is displayed.

2. In the **Schedule** pane, click **Add**.

The **Discovery Schedule** notebook displayed.

3. In the **Name** field, type the name of the discovery schedule.
4. In the **Start Time (server time)** field, type the date and time when you want the discovery schedule to start.
5. From the **Repeat** list, select the frequency that you want the discovery schedule to run.
6. In the **Every** field, type the numeric value for the time interval.
7. To configure the scope of the discovery schedule, click the **Scope** tab. The **Scope** page is displayed.
8. Select the scope elements that you want to include. If you use TADDM 7.3.0.3, or later, select one of the following options:
  - **Fix Pack 3 Dynamic content of selected scopes and groups.** In this mode, you can select only scope sets and scope groups. They are resolved to a list of elements just before a discovery. It means that you can modify the content of such scope sets and scope groups, and the defined schedule runs discoveries with up-to-date list of scope elements. As a result, you do not need to modify a schedule each time you change the content of a scope set, or a scope group.
  - **Fix Pack 3 Static, selected elements of scopes and groups.** In this mode, you can select scope sets, scope groups, and single scope elements. The content of such scope is static, which means that only the chosen elements are discovered. If the scope set, or scope group content changes over time, the discovery is run against the elements that belonged to the scope at the time of the discovery schedule creation.
9. From the **Profile** list, select one of the following options:
  - To discover active computer systems in the runtime environment, select **Level 1 Discovery**. This profile can be used to perform credential-less discovery.
  - To discover detailed information about the active computer systems in the runtime environment, select **Level 2 Discovery**.
  - To discover the entire application infrastructure, deployed software components, physical servers, network devices, virtual LAN, and host data, select **Level 3 Discovery**.
10. To save the discovery schedule, click **OK**.

### ***Viewing discovery schedule details***

You can display the summary information about a discovery schedule on the **Discovery Schedule Details** window.

To display the details for a discovery schedule, complete the following steps from the Product Discovery Management Console:

1. In the Functions pane, click **Discovery > Schedule**.  
The **Schedule** pane is displayed.
2. In the **Schedule** pane, select the schedule that you want to view details for and click **Details**.  
The **Schedule Details** window is displayed. You cannot change any of the details for the discovery schedule. To see what kind of details are displayed, go to [“Schedule Details window” on page 128](#).
3. To close the **Schedule Details** window, click **Close**.

### ***Deleting a discovery schedule***

You can delete an existing discovery schedule.

To delete a discovery schedule, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Schedule**.  
The **Schedule** pane is displayed.
2. In the **Schedule** pane, select the schedule that you want to delete and click **Delete**.  
A message window is displayed.
3. To delete the schedule, click **Yes** in the message window.
4. To confirm the deletion, ensure that the schedule is not listed in the **Schedule** pane.

## Viewing the discovery history

Each time that a discovery is run, the Discovery Management Console updates the discovery activity and error information that is displayed in the **History** pane.

You can view the discovery history, including the associated activity and error information, in the **History** pane. By default, information about the last ten discoveries is displayed.

It might take a long time to retrieve and display the discovery history in the History pane. As an alternative, consider using the Sensor Done Events by Run BIRT report.

The following table lists and describes the information that is displayed for each discovery.

Table 9. Discovery history information	
Field	Description
Start Time	The date and time when the discovery started.
Completion Time	The date and time when the discovery completed.
Completion Code	The final status of the discovery.
Profile Used	<p>The type of profile that the discovery used. This can be one of the following options:</p> <ul style="list-style-type: none"><li>• Level 1 Discovery</li><li>• Level 2 Discovery</li><li>• Level 3 Discovery</li><li>• Custom</li></ul> <p>See <a href="#">“Using discovery profiles” on page 50</a> for more information about discovery profiles.</p>

To view a discovery history, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > History**.  
The **History** pane is displayed.
2. To display information about a discovery, select an entry in the table.  
A second table of data is displayed. This table provides a list of sensors and the host name, IP address, date, status, and description for each sensor.
3. To display the scopes that are included in the discovery, click **Scope Details**. The **Scope List** window is displayed.
4. To close the **Scope List** window, click **Cancel**.

## Using discovery profiles

Discovery profiles help you discover your IT environment.

TADDM discovers and collects configuration information for the entire application infrastructure, identifying deployed software components, physical servers, network devices, virtual LAN, and host data used in a datacenter environment.

For example, you configure individual sensors, manage multiple configurations of the same sensor, pick the appropriate configuration based on a set of criteria, and manage sets of configuration of different sensors to be applied on a single run. You can also specify a discovery profile's access list, and it is used only during a discovery with this particular profile. A discovery profile access list works the same as a general access list.

## Creating discovery profiles

When creating discovery profiles, default profiles, default sensors, and default sensor configurations are not editable.

When you run a discovery, you must select a profile. If no profile is selected, the discovery runs against the default profile, which is Level 3 discovery. To change the default profile, click **Edit > Preferences** and select another profile.

To create discovery profiles, complete the following steps:

1. In the **Discovery** drawer of the Discovery Management Console, click **Discovery Profiles**.
2. In the **Discovery Profiles** window, click **New**.
3. Type the profile name.  
The profile name must be unique.
4. Type a description for the new profile. The description is displayed on the user interface with the **Sensor Configuration**, **Access Control** and **Platform Properties** pages.
5. When you create a new profile, you can use an existing profile as a basis for the new profile. From the **Clone existing profile** list, select an existing profile or select *None*. Cloning an existing profile includes the agent configuration, access list, and platform configuration.

There are three levels of discovery profiles to choose from:

### Level 1 Discovery

This profile can be used to perform credential-less discovery. It can be used to discover active computer systems in the runtime environment.

### Level 2 Discovery

This profile can be used to discover detailed information about the active computer systems in the runtime environment.

### Level 3 Discovery

This profile can be used to discover the entire application infrastructure, deployed software components, physical servers, network devices, virtual LAN, and host data used in a runtime environment. If you are running a discovery using any of the Level 3 layer (application) sensors, computer system sensors from corresponding platforms must be enabled. For example, the Microsoft SQL Server sensor or Citrix Server sensor require the Windows computer system sensor to be enabled during discovery. If you run the application sensor without enabling the computer system sensor, this can cause an application sensor storage error.

6. Click **OK**.

The discovery profile is created and listed with the other existing profiles. The profiles are listed beside the **Sensor Configuration**, **Access Control**, and **Platform Properties** pages. If you cannot see the profiles, look for a splitter bar beside **Sensor Configuration** page. Use your mouse to move the splitter bar to see the list of profiles.

When you select a profile, the details for the profile are displayed on the **Sensor Configuration**, **Access Control**, and **Platform Properties** pages.

7. On the **Sensor Configuration** page, select a sensor and you can create, enable, and configure sensors.

When you configure a sensor, you must first make a copy of the default sensor that is part of the TADDM product. Then, modify that copy. To make a copy of the default sensor, complete the following steps:

- a. Highlight the sensor that you want to modify, and click **New**. The **Create Configuration** dialog is displayed.
- b. Name your new sensor.  
**Note:** When TADDM uses the specified name for file system access, for example logs, static scripts contexts, and so on, all special and whitespace characters are removed from the name.
- c. To enable your configuration and disable the default configuration, click the radio button. Now, you can modify the configuration of the sensor.

You can add scope restrictions to a sensor. A scope restriction means that when a discovery is performed using a profile, the sensor runs only on the scope configured with this scope restriction, that is selected scope sets and scope groups. For example, if you want the *WebSphereSensor* for the *ProfileTest* profile to run on the *WebSphereDiscovery* scope set, create a new sensor configuration based on the WebSphere® cell sensor and configure a scope restriction of *WebSphereDiscovery*. When you run the discovery using the *ProfileTest* profile, select the appropriate scope sets (including *WebSphereDiscovery*) and the WebSphere cell sensor runs only on the WebSphere discovery scope set.

8. On the **Access Control** page, you can add, edit, or delete access control entries for the discovery profile.

Any access controls you set in the **Access Control** page override the access controls in the main access list.

9. On the **Platform Properties** page, you can add, edit, or delete properties for a platform.
10. Click **Save**.

### ***Changing discovery profiles***

Use the **Discovery** drawer of the Discovery Management Console to change discovery profiles.

To change discovery profiles, complete the following steps:

1. In the **Discovery** drawer of the Discovery Management Console, click **Discovery Profiles**.
2. In the **Discovery Profiles** window, select the profile you want to change.

The profiles are listed beside the **Sensor Configuration**, **Access Control**, and **Platform Properties** pages. If you cannot see the profiles, look for a splitter bar beside **Sensor Configuration** page. Use your mouse to move the splitter bar to see the list of profiles.

When you select a profile, the details for the profile are displayed on the **Sensor Configuration**, **Access Control**, and **Platform Properties** pages.

3. On the **Sensor Configuration** page, select a sensor and you can create, enable, configure, and delete sensors.

When you configure a sensor, double-click the value that you want to edit. When you delete sensors, you cannot delete default sensors.

**Important:** To save the edited value, click **Enter**. If you click **OK** after you edit the value, the change is not saved.

4. On the **Access Control** page, you can add, edit, or delete access control entries for the discovery profile.

Any access controls you set in the **Access Control** page override the access controls in the main access list.

5. On the **Platform Properties** page, you can add, edit, or delete properties for a platform.
6. Click **Save**.

### ***Deleting discovery profiles***

Use the **Discovery** drawer of the Discovery Management Console to delete discovery profiles.

To delete a discovery profile, complete the following steps:

1. In the **Discovery** drawer of the Discovery Management Console, click **Discovery Profiles**.
2. In the **Discovery Profiles** window, select the profile that you want to delete. You cannot delete a default profile.

The profiles are listed beside the **Sensor Configuration**, **Access Control**, and **Platform Properties** pages. If you cannot see the profiles, look for a splitter bar beside **Sensor Configuration** page. Use your mouse to move the splitter bar to see the list of profiles.

3. Click **Delete**.

A confirmation message is displayed.



## ***Scheduling discovery profiles***

Use the **Discovery** drawer of the Discovery Management Console to schedule discovery profiles.

To create a schedule for a discovery profile, complete the following steps:

1. In the **Discovery** drawer of the Discovery Management Console, click **Schedule**.
2. In the **Schedule** window, click **Add**.
3. On the **Details** page, complete the following steps:
  - a. Type a name.
  - b. Select a start date and time.
  - c. Select the option for repeating the discovery.
4. On the **Scope** page, complete the following steps:
  - a. Select a scope.
  - b. For the scope, set the options.
  - c. Select a profile.
5. Click **OK**.

## ***Running a discovery using profiles***

Use the **Discovery** drawer of the Discovery Management Console to run a discovery using profiles.

To run a discovery using a profile, complete the following steps:

1. In the **Discovery** drawer of the Discovery Management Console, click **Overview**.
2. In the **Overview** window, click **Run Discovery**.
3. Select the scope elements and components.
4. Select a profile.
5. Click **OK**.

## **The bulk load program**

The bulk load program, which is the `loadidml.sh` file on UNIX systems and the `loadidml.bat` file on Windows systems, loads Discovery Library books into the TADDM database.

A book is a file, in IdML format, that contains data from other Tivoli products. You can load the information in a book into the TADDM database. The Tivoli collection of books is available at <http://www.ibm.com/software/brandcatalog/ismlibrary/>.

The bulk load program reads the books, imports the data into the TADDM database, and logs the results in the `results` directory for the bulk load program. In addition, the bulk load program logs error messages in the `$COLLATION_HOME/log/bulkloader.log` file.

The bulk load program runs on the following types of TADDM server:

- domain server in a domain server deployment
- synchronization server in a synchronization server deployment
- storage server (primary or secondary) in a streaming server deployment

Although the bulk load program is available on a discovery server in a streaming server deployment, it does not run on a discovery server. To ensure proper authorizations, the bulk load program must be run by the same user ID that runs the TADDM server processes.

All the directories that you use to store log and result files must exist prior to running the bulk load program. You can customize these directories by updating the configuration settings in the `$COLLATION_HOME/etc/bulkloader.properties` file.

When the bulk load program completes, the TADDM server might still be processing the IdML book.

When you load data to any object of the LogicalContent class or its descendants, the allowed size for the content attribute is 5 MB. If data exceeds the allowed limit, it is cut and an ellipsis (...) is added in the end of the sentence.

See also “Loading grouping patterns” on page 194, and the *IBM Discovery Library IdML Certification tool* topic in the *TADDM Discovery Library Adapter Developer's Guide*.

### **Running the bulk load program**

The bulk load program provides the capability of loading or updating large amounts of configuration item (CI) data and relationships data into the TADDM database. The input to the bulk load program is a file that contains an Identity Markup Language (IdML) formatted XML document. The bulk load program can also be used to define large number of extended attributes.

In a streaming server deployment, the bulk load program updates data in the database on the storage server. You can run the bulk load program from the primary storage server, secondary storage server, or both at the same time. In a synchronization server deployment, you can run the program from the synchronization server.

To run the bulk load program, complete the following steps:

1. Check the `$COLLATION_HOME/etc/bulkload.properties` file for accuracy.  
To accept the defaults, do not change anything in the file.
2. Verify that the working directory and the results directory mentioned in the `bulkload.properties` file are valid.

The working and results directory must exist before running the bulk load program or the bulk load program does not run. If you want to use different directories, you must create these directories manually and update the properties file. The bulk load program does not automatically create these directories.

To create the directories, use the same user account that starts and stops the TADDM server. If the bulk load program does not have permissions to read and write from the working and results directories, it cannot run.

3. Run the bulk load program.
  - For Windows operating systems, the bulkload script is located in the `$COLLATION_HOME/bin/loadidml.bat` file.
  - For all other operating systems, the bulkload script is located in the `$COLLATION_HOME/bin/loadidml.sh` file.

Use the following command to run the bulk load program:

```
./loadidml.sh -o -f path_to_idml_file
```

Where:

**-o**

Instructs the bulk load program to override the processed files and load the IdML files.

**-f path\_to\_idml\_file**

Specifies the fully qualified path to the input file or a directory that contains input IdML files. The directory where the input file is placed must not be the same as the working directory of the bulk load program. If a shared directory is used to stage the input file, or, if files are copied to a local directory, this directory cannot be the same as the working, results, or log directory of the bulk load program. This parameter is required.

For example,

```
./loadidml.sh -o -f /opt/IBM/taddm/dlaxmls/testfile.xml
```

4. If the bulk load program does not run, read the messages in the `bulkload.log` file. The log file is located in the `$COLLATION_HOME/log` directory.

Depending on the size of the book, the capacity of the computer, and other variables, it might take a long time to load the data. The bulk load program might not write messages to the log file when waiting for the TADDM system to store information in the database. When one or more records are stored in the database, the results file and the log file are updated with the status. You must not cancel the bulk load program while it is loading data. The bulk load program exits when data loading is complete. For information about how to determine whether the bulk load program is running, see the *Bulk load program problems* topic in the *TADDM Troubleshooting Guide*.

5. After the bulk load program runs, check the results file for problems during the bulk load program. The results file is located in the `resultsdir` directory configured in the `bulkload.properties` file. Look for a file with a `.results` extension and named the same as the IdML file. For example, if the name of the imported IdML file is `test.xml`, the name of the results file is `test.results`. If the results file is empty, check for an error in the log file. Important entries in the results file are marked with `SUCCESS` and `FAILURE` tags. If statistics are enabled, percentage successful messages are also recorded. `FAILURE` tags are for individual objects and do not necessarily indicate a failure of the entire file. Objects marked as failed are not stored in the database.
6. To process the same book again after the first initial load, either use the `-o` flag, or remove the specific entry from the `processedfiles.list` file. The `processedfiles.list` file is located in the working directory specified in the `bulkload.properties` file.
7. If the bulk load program indicates that another bulk load program is running and this is not the case, go to the working directory and delete the `.bblock` file. Run the bulk load program again. The `.bblock` file is a hidden file on UNIX systems because it starts with a period (`.`). Delete this file only if you are sure that another bulk load program is not already running. Read the information in the `bulkload.log` file. The log file can contain details about messages that are displayed.
8. You can run the `bulkload` on the synchronization server, however, the following limitations exist:
  - No change propagation: There is a change history similar to that on the domain server, however, there is no change propagation. For example, if you change the duplex setting on an L2 interface, the duplex setting is not displayed as a change to the computer system. The duplex setting is displayed only as a change to the L2 interface.
  - No change aggregation: When an attribute changes from A -> B -> A in the same discovery (`bulkload`), the change (A -> B) or (B -> A) is not recorded in the change history report.
  - Limited advanced reconciliation: The only topology builder agent that runs on the synchronization server is the `CrossDomainDependencyAgent`. If the logical connection has the same IP address for the 'from' and 'to' IP addresses, or the localhost is used, the `CrossDomainDependencyAgent` does not create a dependency. The Discovery Library Adapter (DLA) creates the relationships between both implicit and explicit objects.

For most situations, using just the `-f` and `-o` parameters is sufficient, but other parameters are supported, if needed. The following example shows some infrequently used parameters:

```
./loadidml.sh -f path_to_idml_file -u userid -p passwd  
-g -c -e -o -b bidirectional_format_on_or_auto -l location tag -loadEAMeta  
-override -disableIdmlCertificationTool
```

Where:

**-u *userid***

Specifies the user ID to be used to authenticate with the TADDM server.

The `-u` parameter is optional. Supply a user ID only if the user ID has the correct permissions (full update and read privileges) and is defined in the TADDM server as a valid user.

**-p *passwd***

Specifies the password used to authenticate with the TADDM server.

The `-p` parameter is optional. Supply a password only if the user ID has the correct permissions (full update and read privileges) and is defined in the TADDM server as a valid user.

**-g**

Specifies to use the graph writing algorithm to persist data into the database.

This option improves loading performance, and it is useful for loading XML files with data that has large arrays of contained objects. Tivoli Storage Productivity Center and Tivoli Configuration Manager discovery library IdML files are examples of file with large arrays of contained objects. Other files can also benefit from this algorithm. The graph writing algorithm writes batches of objects to the database at one time. The number of objects written is influenced by the cache size setting in the `bulkload.properties` file. Give careful consideration to the use of this algorithm because there are limitations.

**Restriction:** Because of current API limitations, the IdML file must have source tokens present for each object in order to perform graph writing. Source tokens, however, are an optional value in an IdML XML file. Therefore, if the `-g` option is provided and no source token is available for an object, a dummy source token is automatically generated for that object using the required object ID from the XML file. The dummy source tokens are not displayed as launch in context tokens. However, the dummy source tokens are displayed for individual object attributes and in the bulk load log file. This behavior is a normal part of the algorithm.

If any single element does not satisfy naming rules, or it fails to be written to the database for any reason, the entire graph, or a subset of elements might fail to be persisted. Error messages indicating the specific object that caused the failure are not available due to current limitations. Run the file without the `-g` option to pinpoint a problem.

Certain IdML files reuse source token values for more than one object. While permissible in IdML, these files cannot be processed with the `-g` option due to current limitations. Files that reuse source tokens between objects must be loaded without the `-g` option.

Graph writing requires additional memory at both the client and the server. If an "out of memory" error occurs, reduce the cache size setting in the properties file or do not use the `-g` option.

Abstract resources are not supported during graph writing. Process the files that contain these characteristics without the `-g` option. Extended attributes are supported during graph writing.

**-c**

Specified to copy the IdML source files to the working bulk directory and process them there. This method might lead to delays when copying large files.

**-e**

Specifies that data loading error information is made available in the program return code. By default, the bulk load program returns exit code 0 even if an error occurs when loading data. The `-e` parameter instructs the program to return code 5 when an error occurs when loading data. Note the return code from the bulk load program itself takes precedence even if the `-e` parameter is specified. For example, if the bulk load program cannot connect to the TADDM server, the returned code contains this information.

**-b *bidirectional\_format\_on\_or\_auto***

Specifies whether bidirectional support is enabled, disabled, or automatically configured. Choices for the bidirectional flag are *on* and *auto*. When the bidirectional flag is *on*, you can configure the bidirectional parameters for each Management Software System using the predefined bidirectional profiles. When the bidirectional flag is set to *auto*, the bidirectional transformation is enabled and the bidirectional format is detected automatically.

If you are using SSH, do not specify *on* for the bidirectional flag. When you choose *on* for the bidirectional flag and use SSH, the bulk load bidirectional configuration window is not displayed. Without completing the fields in the bulk load bidirectional configuration window, you cannot configure the bidirectional parameters.

**-l *location tag***

Specifies a location tag value when loading IdML files. Every configuration item that is loaded from the IdML file has this location tag value assigned. If more than one IdML file is present in the same directory and each file requires a unique location tag, you must load the files separately. Make sure

that the `com.ibm.cdb.locationTaggingEnabled` value in the `COLLATION_HOME/etc/collation.properties` file is set to `true`.

For more information about location tagging, see the *Configuring location tagging* topic in the TADDM Administrator's Guide.

### **-loadEAMeta**

**Note:** This flag is related to extended attributes metadata.

Forces bulk loader to ignore values and to store the extended attributes metadata only. New attribute is added to previously defined attributes for the same CDM class in the metadata. The type for the new attribute in attribute metadata is set to 'String'.

If `-loadEAMeta` is passed, the extended attributes metadata can be defined with the following books:

- Regular IdML books
- IdML books with metadata definitions only.

If both are passed, the `-loadEAMeta` option takes precedence over the `-g` option, and the graph writing mode is ignored.

### **Example**

For the following part of the IdML source file, the bulk load program with the `-loadEAMeta` option defines the `myExtAttr1`, `myExtAttr2` and `myExtAttrInCategory` extended attributes for the `WindowsComputerSystem` component type. The `myExtAttrInCategory` attribute is defined in the `myExtAttrCategory` category.

```
<cdm:sys.windows.WindowsComputerSystem id="9.10.10.10-WindowsSystem"
sourceToken="ip_address=9.10.10.10">
  <cdm:extension>
    <cdm:extattr name="myExtAttr1">value1</cdm:extattr>
    <cdm:extattr name="myExtAttr2">value2</cdm:extattr>
    <cdm:extattr category="myExtAttrCategory"
name="myExtAttrInCategory">value3</cdm:extattr>
  </cdm:extension>
  ...
</cdm:sys.windows.WindowsComputerSystem>
```

### **-override**

**Note:** This flag is related to extended attributes metadata.

If this flag is passed with `-loadEAMeta` flag, it forces redefinition of the attribute type, in case when the attribute is already defined, and its type is other than 'String'.

### **Example**

```
./loadidml.sh -f /opt/IBM/taddm/dlaxmls/testfile.xml
-u admin -p password -g -c -o -b auto -l tag
```

### **-disableIdmlCertificationTool**

Specifies to disable IdML books validation before the processing of the books by the bulk load program.

## ***The bulk load properties file***

The bulk load properties file is located in the `$COLLATION_HOME/etc/bulkload.properties` directory. This file gives the bulk load program information that is required to load the IdML file into the TADDM database.

The following list describes the properties in the `$COLLATION_HOME/etc/bulkload.properties` file and its default values. You must not change anything in the file if you want to accept the defaults.

### **com.ibm.cdb.bulk.numcopies=1**

This property specifies the number of copies of the file to copy.

**com.ibm.cdb.bulk.workdir=bulk**

This property specifies the directory the bulk load program uses to copy files to before loading them. See the "-c" option and com.ibm.cdb.bulk.createworkingcopy property. The default directory is relative to the top-level directory of the directory that the \$COLLATION\_HOME environment variable references.

Do not copy the IdML file to the \$COLLATION\_HOME directory, as this location causes the load process to fail.

**com.ibm.cdb.bulk.workdir.cleanup=false**

This property specifies whether the working directory is cleaned up after the load process is finished.

**com.ibm.cdb.bulk.processedfiles.cleanup=30**

This property specifies the number of days to keep files in the processed files list.

**com.ibm.cdb.bulk.retrycount=5**

This property specifies the number of times to try loading a file again if the allowed number of concurrent bulk loads, which is 10 by default, is exceeded.

**com.ibm.cdb.bulk.retrydelay=60**

This property specifies the number of seconds to wait before trying to load a file again, while a discovery is in progress.

**com.ibm.cdb.bulk.resultsdir=bulk/results**

This property specifies the directory to place the results files created, when an IdML file is loaded into the TADDM database. The default directory is relative to the top-level directory referenced by the \$COLLATION\_HOME variable.

**com.ibm.cdb.bulk.apiservertimeout=60**

This property specifies the number of seconds before the API server returns an error and the bulk load program stops processing.

**com.ibm.cdb.bulk.stats.enabled=false**

This property specifies whether statistics gathering of the bulk load program are performed. Turning on statistics decreases performance and increases log and result file sizes.

**com.ibm.cdb.bulk.log.success.results=true**

This property specifies whether successfully written objects are logged to the results file. Reduced logging can improve performance by reducing output.

**com.ibm.cdb.bulk.allocpoolsize=1024**

This property specifies the maximum amount of memory that can be allocated to the Bulk Loader process. It is an Xmx value that is passed to the main Java class of the Bulk Loader. Specify the value in megabytes.

**com.ibm.cdb.bulk.cachesize=2000**

This property specifies the number of objects to be processed in a single write operation when performing graph writing. Increasing this number improves performance at the risk of running out of memory either on the client or at the server. Alter this number only when specific information is available to indicate that processing a file with a larger cache provides a benefit in performance. The cache size setting currently can be no larger than 40000.

**com.ibm.cdb.bulk.createworkingcopy=true**

This property specifies to first copy the IdML source file to the bulk directory and then continue to process the copied file.

**com.ibm.cdb.bulk.forceUniqueSourceTokens=true**

This property specifies whether unique source tokens are created by the bulk loader. The default value is true. To disable the creation of unique source tokens, set the value to false.

This property is used only if graph writing is enabled.

When using graph writing, lower level objects with duplicate source tokens might not be displayed correctly when launched in context. When this property is set to true, an index number is appended to a duplicate source token to ensure uniqueness.

**com.ibm.cdb.bulk.idmlcertificationtool.disabled=true**

This property specifies whether IdML Certification Tool is used to validate books before their processing. By default, the certification tool is disabled.

**com.ibm.cdb.bulk.idmlcertificationtool.toolongattrhandling=error**

This property specifies how IdML Certification Tool handles too long CDM attribute values. The following values are supported:

- **error** - the default value, reports too long CDM attribute values as errors.
- **warn** - reports too long CDM attribute values as warnings.
- **ignore** - ignores problems with too long CDM attribute values.

By default, too long attribute values are reported as errors and IdML books processing is stopped with a parsing error. When the property is set to **warn** or **ignore**, the bulk loader program is not stopped.

It is advisable not to ignore the problem with too long attribute values. Although too long attribute values are truncated and stored in the TADDM database, they can cause information inconsistency. It can lead to an unexpected behavior of TADDM and products that are integrated with TADDM.

This property is used only when IdML Certification Tool is enabled.

***Bulk load return codes***

The following return codes are set so that if you are writing a cron script or some other script that calls the bulk load program, you can determine the status of the bulk load and how the bulk load exited.

At a command line, you see the following return codes and their messages.

**0**

The program completed. This does not mean that everything was loaded. Check the results file for that information.

**1**

Some error occurred but it is unknown. Check the `bulkload.log` file in the `log` directory to see if there is more information.

**2**

A basic environment property needed to run the bulk load program is not set.

**3**

A command line parameter was supplied that is not valid. It is either the parameter itself or the data supplied with the parameter that is not correct. Correct the command and try again.

**4**

The user ID or password was not correct and the bulk load program could not log in. This happens when an incorrect `-u -p` parameter was supplied to the bulk load program.

**5**

The XML file being processed contained errors but the bulk load program continued to process the file.

**6**

The XML file being processed contained errors and caused the bulk load program to stop the processing of the file.

**7**

The XML parser failed to parse the XML file and the bulk load program processing stopped.

**8**

The API server returned an error but the bulk load program was able to recover and continue.

**9**

The API server returned an error and the bulk load program stopped processing.

**10**

Only one copy of the bulk load program can run at a time. A copy was already running so this copy can not run.

**11**

A discovery is processing and the bulk load program is locked out and can not run. Based on what is configured in the properties file, the bulk load program tries to run again, but if this error is returned, it has exhausted the retry attempts.

**12**

A discovery is processing and the bulk load program is locked out and can not run. Based on what is configured in the properties file, the bulk load program tries to run again, but if this error is returned, it has exhausted the retry attempts.

**13**

There is an property specified in the input file for the bulk load program that is not valid.

**14**

The file was already processed as recorded in the `processedfiles.list` file in the working directory of the bulk load program. Either use the `-o` override parameter to force processing of the file or edit the `processedfile.list` and remove the entry for this file from the list.

**15**

The API server was not started and the bulk load program could not connect.

**16**

The properties files for bidirectional languages were empty.

### ***Best practices for using the bulk load program***

You should load multiple input files in the correct order (according to time or alphabetically), refresh your latest files, and remove files from your shared directory before they expire from the processed list file.

There are two approaches to control the order in which multiple input files from a directory are loaded. One option is to load each file individually, loading the files in the correct order. The approach might be necessary if the only difference between the file names is a time stamp in the file name. A second option is to change the names of the files to include alphabetic ordering strings. These ordering strings are then defined to the bulk load program using the `processOrder.list` file. The `processOrder.list` file does not exist, therefore you must create it manually. The bulk load program processes files that match the first ordering string first, the second ordering string second and so on. If more than one file matches the same ordering string, a processing order within that group is not ensured.

For refresh files, typically only the latest refresh file should be loaded. For refresh and delta files, the refresh file should typically be loaded first and then the delta files are loaded in the sequence in which they were generated. For just delta files, they should be loaded in the sequence in which they were generated.

A shared directory used for input files must be properly maintained. Loaded input files must be removed from the shared directory before they are expired from the processed list file. If a file remains in the directory after being expired from the processed list, it is reloaded, perhaps loading older data.

### **Frequent performance of database tuning during the initial DLA books loads**

TADDM requires database maintenance to optimize resource usage and improve performance of SQL queries. The TADDM bulk load program (`loadidm1.sh` file for UNIX systems or `loadidm1.bat` for Windows systems) reads and updates database objects while moving the data from DLA books into the TADDM database. The successful maintenance requires:

- Loading the representative data into the database that is later used for statistics calculation.
- Performing TADDM database maintenance according to the *Database maintenance* topic in the TADDM *Administrator's Guide*, to gather database statistics.

Performing database maintenance frequently during the initial DLA books loads significantly improves performance and reduces the time that it takes SQL queries (SELECT, UPDATE, DELETE) to run against the database. After the initial DLA books loads, this process is not required because the database statistics are valid.

#### **Performing database maintenance during the initial DLA books loads.**

The following example refers to the z/OS DLA books.



1. Load each type of a book from one of your smaller LPARs (BASE, TASK, DB2, IMS, CICS, ZOS, MQ, WAS).
2. Perform database maintenance to gather the database statistics. For DB2, you must run RUNSTATS/REORG statements.
3. Load all the books from one of your smaller LPARs (BASE, TASK, DB2, IMS, CICS, ZOS, MQ, WAS).
4. Perform database maintenance again to update the database statistics.
5. Load all the books from one of your largest LPARs (BASE, TASK, DB2, IMS, CICS, ZOS, MQ, WAS).
6. Perform database maintenance again to update the database statistics.
7. Load the rest of the books. Perform database maintenance online during the bulk load process every couple of hours to gather the latest database statistics.

## Using Bidirectional (BiDi) support for sensors and the bulk load program

Bidirectional language support is provided for Arabic and Hebrew languages.

BiDi text is stored and processed in different environments (platforms) and with different layouts. This BiDi text can be introduced in TADDM through sensor discovery, IdML books, Bulk loading, or APIs. Bidirectional *layout transformation* must be used to transform from an external layout format to the TADDM default BiDi layout format.

### ***Bidi support for sensors***

To deal with data that is discovered and contains BiDi data with different formats, you have to transform the BiDi format of all the discovered BiDi data to a BiDi Default Format. Following the Unicode standards, the BiDi Default Format parameters values are:

- Text Type: Implicit
- Text Direction: Left-to-Right (LTR)
- Symmetric swapping flag: Yes
- BiDi Shaping indicator: Not Shaped
- BiDi Numeric indicator: Nominal

You can enable or disable BiDi Transformation for the data sensors.

The BiDi format is configured in one of the following ways:

- Configuration of BiDi format used by external system prior to discovery. You must explicitly specify this method BiDi format prior to a discovery run.
- Automatic discovery of BiDi format standard used by external system during discovery. This method BiDi format is identified based on data attributes and external system attributes or both, that are discovered by the sensor. The identification of BiDi format is based on the algorithm taking into account some or all of the information or data that is discovered by the sensor from the external system.

During discovery configuration you can specify which of these two format options that you prefer. If you prefer the first one, it is also possible to specify the BiDi Profile to be used during discovery.

### ***Bidi support for the bulk load program***

Discovery adapters create the IdML files in the Discovery library. The Discovery Adapters write data into the Discovery Library. The discovered data can contain BiDi data. In this case, a BiDi transformation is needed for this data to transform the data to the default BiDi format.

The BiDi support for the IdML data is provided in the Bulk Load program. The Bulk Load program configures the needed BiDi format for the loaded data from the IdML files depending on the User BiDi configuration. You can configure BiDi format for each Management Software System or use an automatic BiDi format detection option. See [“Configuring bidirectional format for the bulk load program”](#) on page 63 for more information.

## Creating a bidirectional profile

Bidirectional language support is provided for Arabic and Hebrew languages.

If you need to configure BiDi for sensors or for the Bulk Load program, you should create a Bidirectional profile. You use the BiDi profile to do the following things:

- Assign a name for each specific BiDi format. The following list includes the attributes you can select for each profile:
  - Text type
  - Direction
  - Symmetric swapping
  - Shaping
  - Numerical shaping
- Reuse the BiDi Profile in several configurations; sensor configuration and bulk load program configurations.
- You can select different BiDi attributes and any change to these attributes is reflected in the sensor configuration or the bulk load program.

To create a bidi profile, complete the following steps:

1. Log in with a user ID that gives you update permission.
2. On the menu bar, click **Edit > Bidi Profiles**.
3. To add a new bidirectional profile configuration, click **Add**.
4. In the **Bidi Profile** pane, complete the following fields:
  - a) Enter a profile name. The profile name can contain spaces and supports NLS characters.
  - b) Optional: Enter a brief description for this profile.
  - c) Select the BiDi attributes from the drop down lists. Use the following table to help with your selections:

Table 10. BiDi attributes				
Parameter	Values	Description	Default Setting	Comment
Text Type	I	Implicit (Logical)	I (Implicit)	
	V	Visual		
Text Direction	L	Left to Right	L (Left to Right)	
	R	Right to Left		
	C	Contextual Left to Right		
	D	Contentual Right to Left		
Symmetric Swapping	Y	Symmetric swapping in on	Y (Yes)	
	N	Symmetric swapping is off		
Shaping	S	Text is shaped	N (Not shaped)	Applicable for Arabic scripts only
	N	Text is not shaped		
	I	Initial shaping		
	M	Middle shaping		
	F	Final shaping		
	B	Isolated shaping		

Table 10. BiDi attributes (continued)				
Parameter	Values	Description	Default Setting	Comment
Numeric Shaping	H	Hindi (National)	N (Nominal)	Applicable for Arabic scripts only
	C	Contextual		
	N	Nominal		

5. Click **OK** to save the new BiDi profile.

### **Configuring bidirectional format for discovery**

To display data that is discovered with bidirectional data in different bidirectional formats, the TADDM discovery process transforms the bidirectional data into one format.

Configuring the bidirectional format for discovery is a two-step process: select a sensor and set the configuration option.

To configure bidirectional format for discovery, complete the following steps:

1. Log in with a user ID that gives you update permission.
2. On the menu bar, click **Edit > Bidi Configuration**.
3. To add a new bidirectional format for discovery, click **Add**.
4. Select a sensor name for discovery.
5. Select one of the three options for configuration:

#### **Bidi transformation OFF**

Support for the bidirectional format is disabled. This option is the default option.

#### **Automatic Bidi discovery**

Support for the bidirectional format is enabled. The bidirectional format is automatically detected.

#### **Bidi transformation ON**

Support for the bidirectional format is enabled. If you select this option, you can select a bidirectional profile from the list.

6. To save the bidirectional format for discovery, click **OK**.

### **Configuring bidirectional format for the bulk load program**

To display data that is discovered with bidirectional data in different bidirectional formats, the TADDM discovery process transforms the bidirectional data into one format.

The **loadidm1.sh** script is used to start and run the Bulk Loader program from a command line on Linux, or AIX systems. The script needs several command line parameters:

**-h hostname**

**-u user id**

**-p password**

**-f path+filename**

**-b on/auto/file**

This option is used to enable and configure BiDi support for the bulk load program, according to the following parameters:

- On: Turn on the BiDi transformation
- Auto: Turn on automatic BiDi transformation
- file: Enable and configure the imported CI using a BiDi configuration file

This parameter is also used to enable and configure BiDi support for the bulk load program:

### On

With this parameter, a BiDi configuration window is displayed where you can configure a BiDi profile for each Management Software System (MSS). You can configure MSS using a pre-defined BiDi Profile or selecting an **Automatic BiDi profile** to turn on automatic BiDi transformation for this MSS. Using `-b ON`, you can open a window to see the Bulk Load BiDi configuration GUI. You are not able to use this option when you use Secure Shell to start it. This option is manual so it can not be used when you want to run the Bulk Load program silently.

### Auto

With this parameter, the BiDi transformation is enabled and the BiDi format is detected automatically without any interaction from you.

### file

With this parameter, BiDi configuration for each MSS is done using a BiDi configuration file. Create this file, `bidiconfig.properties` in the `$COLLATION_HOME/etc` path.

In the bidi configuration file, configure each MSS to a pre-created BiDi Profile: `MSS_Name = BiDI_PROFILE_NAME`. This option requires the MSS and its BiDi configuration to be added to the bidi configuration file in advance. For configuring any MSS using “`-b file`” option which was already loaded into the TADDM database, the profile specified in BiDi configuration file overrides the configuration specified in the TADDM database. The following warning will be printed in the `bulkload.log` file:

```
"BIDI Warning: MSS_NAME has two BiDi configurations.  
The profile specified in properties file will override  
the configuration specified in CMDB"
```

Complete the following steps to configure bidirectional format for the Bulk Load program:

1. Log in with a user ID that gives you update permission.
2. On the menu bar, click **Edit > Bidi Profile** and add a new BiDi profile.
3. Run the `loadidml` script using the BiDi option ‘`-b`’, and pass one of the following parameters to this option:

- Auto

```
loadidml.sh -u administrator -p collation -b auto -f /sampleidml.xml
```

- On

```
loadidml.sh -u administrator -p collation -b on -f /sampleidml.xml
```

- file

```
For example: loadidml.sh -u administrator -p collation -b file  
-f /sampleidml.xml
```

## Reconfiguring BiDi profile for a loaded MSS

You can reconfigure the BiDi profile for a loaded MSS.

Complete the following steps to reconfigure the bidirectional format for a loaded MSS:

1. On the menu bar, click **Edit > MSS**
2. In the **Management Software System List**, select the MSS and click **Edit**.
3. Select one of the following options:
  - Another BiDi profile that was created previously.
  - **Automatic BiDi Profile** to turn on the Automatic BiDi transformation.
  - **Empty** to disable the BiDi Transformation.
4. Click **OK**.

## Delta books utility program

Instead of loading the full DLA book when data changes occur, TADDM provides the possibility to generate and load the delta book only. With the delta books utility program, you can generate the IdML delta book sets that contain delta or changes between two sets of books. In typical situations, the program greatly increases the speed of the book loading process.

The delta books utility program is deployed in the following files:

- For UNIX:

```
$COLLATION_HOME/tools/deltabooks/deltabooks.tar
```

- For Windows:

```
$COLLATION_HOME/tools/deltabooks/deltabooks.zip
```

See also the *IBM Discovery Library IdML Certification tool* topic in the *TADDM Discovery Library Adapter Developer's Guide*

## Loading DLA books into TADDM

You can use the following procedure to load the DLA books into TADDM.

1. Run the z/OS® DLA on a set of targets to gather the DLA books.
2. Load the discovery output into TADDM.
3. After the books have changed, run the z/OS DLA tool again on the same targets.
4. Use the delta books utility to run the first and the second output of the z/OS DLA, gathered in step 1 and 3.
5. Load the delta books into TADDM.

### Related concepts

[“The bulk load program” on page 53](#)

The bulk load program, which is the `loadidml.sh` file on UNIX systems and the `loadidml.bat` file on Windows systems, loads Discovery Library books into the TADDM database.

## Using the delta books utility program

After you gather the DLA books and load them to TADDM, you can use the delta books utility program to generate the delta books.

1. To use the delta books utility program, extract the following archives:

- For UNIX:

```
$COLLATION_HOME/tools/deltabooks/deltabooks.tar
```

- For Windows:

```
$COLLATION_HOME/tools/deltabooks/deltabooks.zip
```

2. Run the `deltabooks.sh` script or the `deltabooks.bat` batch file with the following syntax:

```
deltabooks.sh|bat -f <fully qualified path to directory that contains first discovery output>  
-t <fully qualified path to directory that contains the second discovery output>  
-o <fully qualified path to directory for the delta books> [-verbose] [-allowReversedOrder]
```

where:

- The `-verbose` parameter enables verbose output.
- **Fix Pack 2** The `-allowReversedOrder` parameter allows to reverse the order of books comparison so that the books from the first discovery are newer than the corresponding books from the second discovery.

**Note:** This parameter is available in TADDM 7.3.0.2 and later.

Example: `deltabooks.sh -f /first_output/ -t /second_output/ -o /delta_books/`

3. Review the delta book output to make certain that all book-pairs were processed. Especially look for lines that contain the following messages:

- Failed to generate delta book
- Failed to generate delta books

## Application descriptors

You can use application descriptors to associate components to business applications and specify further details about the applications.

### ***Application descriptors overview***

IBM Application descriptors provide complete automation of the process of discovering, creating, and maintaining business applications and their composition.

An application descriptor is an application tag that maps a computer system, application server, or module to a business application. By using application descriptors, you can identify a component of a business application at development time. When application descriptors are discovered, they are used to automatically associate components with business applications, thus eliminating manual modeling and maintenance of business application compositions.

An application descriptor is an XML file that is placed in a specified location that specifies computer systems, application servers (containers), or modules, and associates them with business applications. You can map multiple modules at one time or map an entire container (such as an IBM WebSphere server).

The `$COLLATION_HOME/log/services/TopologyBuilder.log` and `$COLLATION_HOME/log/agents/AppDescriptorAgent.log` files contain log messages relevant to the processing of application descriptor files.

You can use the following strategies for creating and deploying application descriptors:

#### **During deployment**

Application definition during development and deployment is the recommended approach. With this approach, you can capture the most accurate and complete information about the packaging of modules in business applications.

#### **After deployment**

You can add application descriptors for deployed modules after initial deployment, by creating the descriptors, and then deploying them to the file system on the target computer.

There are two types of application descriptors:

#### **Base application descriptor**

Contains general information about an application. The base application descriptor is optional.

#### **Component application descriptor**

Contains information about a specific computer system, application server, or module deployed within a server.

You must assign a unique application name in both the base application descriptor and the component application descriptor. This unique name is used to correlate all discovered application descriptors for a specific application.

**Restriction:** Application descriptors used for business applications are not supported by the script-based or asynchronous discovery sensors. During the script-based or asynchronous discovery, the application descriptors are not discovered.

## Base application descriptor

The base application descriptor contains general information about a grouping pattern, and, in effect, also about a business application, such as the description, URL, contact, and other information.

Because the base application descriptor contains general information, it is not required to discover an application.

**Important:** Business applications are created automatically even without a base application descriptor file if a component application descriptor file is provided and contains an `app-instance-name` tag. The name that is used for the business application is the `app-instance-name` tag from the component application descriptor file. Moreover, the base application descriptor without any component application descriptor with a matching application name does not trigger the creation of a grouping pattern, and, in result, a business application.

You need only one base application descriptor for each application. In cases when more than one descriptor is used, the system uses the one with the most recent time stamp.

The base application descriptor can be deployed to any descriptor directory of any component of the application.

The following table describes the structure of the base application descriptor:

Table 11. Base Application Descriptor elements and attributes		
Element	Description and attributes	
base-app-descriptor	The root element for the base application descriptor.	
app-instance	The element for the application instance information.	
	name	(Required) The name of the application instance.
	grouping-pattern	The name of the grouping pattern that includes the definition of the business application.
	description	A description of the application instance.
	url	The URL pointing to the application.
	contact	A contact name or other information for the application (This is not imported into TADDM).

The following XML snippet shows an example of the base application descriptor:

```
<base-app-descriptor>
  <app-instance
    name="application_name"
    grouping-pattern="grouping_pattern_name"
    description="application_description"
    url="application_url"
    contact="contact_name"/>
</base-app-descriptor>
```

## Component application descriptor

The component application descriptor contains information about a specific computer system, server, or module deployed within a server, along with information about the participation of the component within the application.

Components can include computer systems, database servers, Java EE servers, or modules within servers. You can use a separate descriptor for each module, or a single descriptor for all modules within a server.

A component application descriptor must be deployed in the descriptor directory of each server that is a component of the business application, or contains modules that are components of the business application. Use component application descriptors instead of the WebSphere or Weblogic sensors to add the Java EE components to a business application for a finer granularity to the discovered dependencies. For more information, see *Best Practices for Discovering Business Applications* on the TADDM wiki at <https://github.com/TADDM/taddm-wiki/wiki/Business-Application-Mapping>

An application descriptor is an XML with the following format:

```
<component-app-descriptor
  app-instance-name="instance_name"
  grouping-pattern>
  <component-descriptor
    type="component_type"
    name="component_name"
    marker-module="true/false"/>
</component-app-descriptor>
```

The elements and attributes of the component application descriptor file are as follows:

Table 12. Component application descriptor elements and attributes		
Component descriptor element	Description and attributes	
component-app-descriptor	The root element for the component application descriptor.	
	app-instance-name	(Required) The name of the application instance.
	grouping-pattern	The name of the grouping pattern that includes the definition of the business application.



Table 12. Component application descriptor elements and attributes (continued)

Component descriptor element	Description and attributes	
component-descriptor	(Required) The element for the component information.	
	type	<p>(Required) A component descriptor can apply to a computer system (host), a server in its entirety, or to individual modules within a server. The type attribute specifies this relationship, and can have either of the following values:</p> <ul style="list-style-type: none"> <li>• host - a computer system.</li> <li>• server - a software server, for example an application server.</li> <li>• module - a software module deployed on a server.</li> <li>• deployable - any component that is deployed on a server.</li> </ul>
	name	The name of the component. Required when the type attribute is set to module or deployable.
	marker-module	<p>(Optional) A special type of module definition for Java EE domains. When a module is indicated as a marker module, Java EE-managed servers within the domain that include the marker module are treated as having all of its modules included in the application. For other types of software servers that are not application servers like Java EE domains, the marker module indicates that all the deployed components on the server that include the marker module are included in the business application. You can specify the following values for the marker module:</p> <ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>

**Note:** To remove a component from a business application that was created with a component application descriptor, you must edit the business application in the Data Management Portal.

### Application descriptor locations

The location of the directory containing application descriptors depends upon the component type and the system configuration.

An application descriptor file is an XML placed in a specific location that depends upon the type of application descriptor. The file name is not significant, but it must end in the .xml extension.

**Note:** Make sure that the TADDM service account has access to the location of any application descriptor you want to use.

During discovery, the TADDM server checks for application descriptors as follows:

- For base application descriptors, the application descriptor file can be placed in the application descriptor directory of any component that is part of the application.
- For computer system (host) application descriptors, the application descriptor directory is specified by the **com.collation.platform.os.hostappdescriptorfiles.dir** parameter in the collation.properties configuration file on the TADDM server, as in the following example:

```
com.collation.platform.os.hostappdescriptorfiles.dir="/home/taddm/hostappdesc"
```

A component application descriptor xml file of type host must contain only the component application descriptor of the same type. Only such files are processed from the directory specified by com.collation.platform.os.hostappdescriptorfiles.dir.

- For CST application descriptors, the directory can only contain:
  - component-app-descriptor with type that is not equal to host
  - base-app-descriptor
- For application server and module descriptors, the application descriptor directory is a subdirectory named appdescriptors in one of the following locations (listed in order of priority):
  1. The custom path that is specified by the COLL\_APP\_DESC\_DIR environment variable. On the target system, set the COLL\_APP\_DESC\_DIR environment variable in the shell prior to starting the application that you want to discover.
  2. The custom path that is specified by the COLL\_APP\_DESC\_DIR command-line argument. On the target system, start the program with a command-line argument COLL\_APP\_DESC\_DIR=path.
  3. The default application descriptor location as specified in [Table 13 on page 70](#), if no custom path is specified.

Table 13. Default Application Descriptor Locations		
Server	Supported modules	Default directory and supported modules
WebSphere Application Server version 6 and later	Java EE applications, web, EJB, and connector modules	<i>WebSphere_profile_dir</i> /appdescriptors,
WebLogic	Java EE applications, web, EJB, and connector modules	<i>WebLogic_home_dir</i> /appdescriptors
JBoss	Java EE applications, web, EJB, and connector modules	<i>JBoss_home_dir</i> /appdescriptors
IPlanet	Servlets, JSP pages	<i>IPlanet_home_dir</i> /appdescriptors
Apache		<i>apache_server_root</i> /appdescriptors
Microsoft IIS	Virtual hosts	<i>IIS_home_dir</i> /appdescriptors
Oracle	Users	<i>Oracle_home_dir</i> /instance_name/appdescriptors <b>Note:</b> You must create the instance_name for this location.
Sybase/Sybase IQ	Databases	<i>Sybase_home_dir</i> /appdescriptors
SQLServer	Databases	<i>SQLServer_home_dir</i> /appdescriptors
DB2®		<i>\$DB2INSTANCEHOME</i> /appdescriptors
Domino® Server		<i>Domino_server_home_dir</i> /appdescriptors
Microsoft Exchange Server 2003	Exchange Servers, Exchange Protocol Virtual Servers	<i>exchange_server_home_dir</i> /appdescriptors
Custom Server	User-supplied through template definition	User-supplied through template definition
Veritas Cluster		<i>VS_home_dir</i> /appdescriptors

In the case of managed servers such as Java EE servers, which are managed by the Java EE domain, the location of the application descriptor directory is at the level of the Admin Server or Domain Manager. The contents that are specified in that directory are used as the superset of all possible mappings for all managed servers. For each managed server (depending on which modules are discovered as deployed), the application descriptor is processed for inclusion of those modules in the application.

## Application descriptors example

You can view application information for a sample application descriptor.

The following table describes details of a sample application:

Table 14. Application description	
	Detail
Application	Order Management
Instance	Staging
Servers	<ul style="list-style-type: none"><li>• Three Apache servers</li><li>• Two WebLogic servers (managed in one domain)</li><li>• One custom server (Order Fulfillment gateway)</li><li>• One custom Java process (Automailer)</li><li>• One Oracle Instance</li></ul>
Modules	<ul style="list-style-type: none"><li>• Static content</li><li>• WAR file</li><li>• EAR file</li><li>• RAR file (communicates with the gateway module)</li><li>• Virtual gateway module</li><li>• automailer jar</li><li>• DB schema</li></ul>
Host	<ul style="list-style-type: none"><li>• Three Web server systems</li><li>• Two application server systems</li><li>• One Order Fulfillment gateway server system</li><li>• One Oracle database server system</li></ul>

The base application descriptor for the sample application is stored in the `oms_coll_desc.xml` file, placed in the application descriptor directory of any application component:

```
<base-app-descriptor>
  <app-instance
    name="Order Management-Staging"
    description="Order Entry application- staging"
    url="http://orderentry.stage.lab.com"
    contact="John Public" />
</base-app-descriptor>
```

The component application descriptor for the computer system hosting the Apache server is stored in the `apache_host_coll_desc.xml` file in the location specified by the **`com.collation.platform.os.hostappdescriptorfiles.dir`** parameter in the `collation.properties` configuration file. Additional host application descriptors are present on each computer system included in the business application:

```
<component-app-descriptor
  grouping-pattern="Order Management-Staging"
  app-instance-name="Order Management-Staging-Web Tier" >
  <component-descriptor
    type="host"
    name="staging.example.com"
    marker-module="true" />
</component-app-descriptor>
```

The WebLogic component application descriptor is stored in the `wls_coll_desc.xml` file in the `WebLogic_home_dir/appdescriptors` directory:

```
<component-app-descriptor
  grouping-pattern="Order Management-Staging"
  app-instance-name="Order Management-Staging-Order Processing Server" >
  <!-- order.war -->
  <component-descriptor
    type="module"
    name="WebLogicWebModule:order"
    marker-module="false" />

  <!-- orderejb.ear -->
  <component-descriptor
    type="module"
    name="WebLogicWebModule:orderejb"
    marker-module="false" />

  <!-- ofg.rar -->
  <component-descriptor
    type="module"
    name="WebLogicWebModule:ofg"
    marker-module="false" />
</component-app-descriptor>
```

The Order Fulfillment Gateway component application descriptor is stored in the `ofg_coll_desc.xml` file in the directory specified in the custom server template:

```
<component-app-descriptor
  grouping-pattern="Order Management-Staging"
  app-instance-name="Order Management-Staging-Order Fulfillment Gateway" >
  <component-descriptor
    type="server"
    name="n/a"
    marker-module="false" />
</component-app-descriptor>
```

The component application descriptor for the automailer Java process is stored in the `am_coll_desc.xml` file in the directory specified in the custom server template:

```
<component-app-descriptor
  grouping-pattern="Order Management-Staging"
  app-instance-name="Order Management-Staging-Order Processing Automailer">
  <component-descriptor
    type="module"
    name="automailer.jar"
    marker-module="false" />
  <component-descriptor
    type="module"
    name="login.jar"
    marker-module="false" />
</component-app-descriptor>
```

The component application descriptor for the Oracle database schema is stored in the `ora_coll_desc.xml` file in the `$ORACLE_HOME/appdescriptors` directory:

```
<component-app-descriptor
  grouping-pattern="Order Management-Staging"
  app-instance-name="Order Management-Staging-Order Processing DB">
  <component-descriptor
    type="module"
    name="ORDER"
    marker-module="false" />

  <component-descriptor
    type="module"
    name="OFG"
    marker-module="false" />

  <component-descriptor
    type="module"
    name="ADMIN"
    marker-module="false" />
</component-app-descriptor>
```

The component application descriptor for the Apache server is stored in the `apache_coll_desc.xml` file in the `Apache_server_home_dir/appdescriptors` directory on the Apache server. Additional application descriptors are present on each Apache server that is part of the Order Management application:

```
<component-app-descriptor
  grouping-pattern="Order Management-Staging"
  app-instance-name="Order Management-Staging-Web Tier">
  <component-descriptor
    type="module"
    name="/opt/apache13/htdocs/ordermgt/"
    marker-module="false" />
  </component-app-descriptor>
```

## Suppressing sensor warnings

You can filter selected warning messages, when the same set of warnings is reported each time the discovery is run.

Sensors produce warning messages when some of the discovery data is not collected. You can filter the messages about the problems that are already known.

**Restriction:** Filtering warning messages is possible only for TADDM warning codes and other language-independent identifiers, such as operating system error codes that constitute a warning message. Localized language-dependent message parts cannot be filtered.

In the `collation.properties` file, add a list of warning messages that you want to filter to the `com.ibm.cdb.discover.suppressedWarnings` property. Such a list consists of sensor warning codes or OS commands that are a part of a warning message. The codes and commands must be separated by semicolons. Begin the list with the CTJTD warning codes.

The filtering settings do not affect IPs and the scopes that have their own scoped version of this property.

- To filter warnings for specific IP or scope, append IP address or scope name to the property:

```
com.ibm.cdb.discover.suppressedWarnings.<IP>=<filter>;<filter>;...
```

```
com.ibm.cdb.discover.suppressedWarnings.<scopename>=<filter>;...
```

- To filter all warning messages, use an asterisk (\*) as one of the filters.
- To include a semicolon in the filter, use the following sequence: \\;

### Examples

**Important:** Type the warning codes and commands in one line.

- The following example shows filtering specified warnings for a target computer system or a discovery scope:

```
com.ibm.cdb.discover.suppressedWarnings.1.2.3.4=CTJTD0808W;CTJTD0737W
com.ibm.cdb.discover.suppressedWarnings.myscope=cat /proc/cpuinfo | grep core id;
CTJTD0762W;0x32bf
```

- The following example shows filtering all warnings for target computer system:

```
com.ibm.cdb.discover.suppressedWarnings.1.2.3.4=*
```

- The following example shows filtering a message that contains a semicolon:

```
com.ibm.cdb.discover.suppressedWarnings=LANG=C\\; psrinfo -p
```

**Note:** You can also specify the `com.ibm.cdb.discover.suppressedWarnings` property via Product Console in Platform Properties window of Discovery Profiles. When the **Included** check box for the `com.ibm.cdb.discover.suppressedWarnings` property is selected, this property overrides the filtering settings from the `collation.properties` file when the discovery is run with this discovery profile.

## Reconciling configuration items

Configuration items (CIs) are reconciled to determine if a newly discovered CI matches a CI that is stored in the TADDM database. This process eliminates duplication in the TADDM database.

Reconciling new data occurs automatically when the following events take place:

- After running a discovery, but before the newly discovered CIs are stored in the TADDM database, the reconciliation process starts. This process eliminates duplicates and cuts down on the overall processing time for discovered CIs. The process is carried out for computer systems discovered by various TADDM sensors and for computer systems loaded from discovery library adapters (DLAs).
- During object storage, the topology manager merges CIs based on matching naming rules.
- Periodically, the topology builder agents run to perform a more complex reconciliation. This method includes the merging of existing CIs, creating relationships, and removing any duplicate CIs not covered by the previous events.

## Prioritization of data

You can use prioritization to order incoming data into the TADDM database. Prioritization uses defined rules to determine which data source takes priority over other data sources when updating configuration item (CI) attributes. This method ensures that the reconciled CIs contain attribute values from a predefined data source.

Data for CIs can be supplied to TADDM from multiple sources. The multiple sources can include various sensors, the Discovery Management Console, the API, or discovery library adapters (DLAs). Prioritization of data, uses rules to create an ordered list of data sources.

## Prerequisites for using prioritization

The following conditions must exist before using prioritization rules:

- Prioritization occurs only when data is written and does not affect existing data.
- Prioritization can only occur when loading data sources into a single database. Cross domain data, such as data combined at the synchronization server, cannot be prioritized.
- Prioritization and data sources rules can be defined either before or after data from a particular source is loaded into the TADDM database. Deleting a data source does not affect data in the database.
- Prioritization can only be applied between two CIs that are recognized by the system as the same item. When the same CI is written to the database using different naming rules or different values for the attribute, the system views the CI as two separate items. Prioritization is not applied to data between two different CIs.

CI data that does not require prioritization does not need a data source definition in order to save data into the TADDM database. In many cases, it is not necessary to provide any data sources or priority rules for a CI. Prioritization is not required for example, if only one data source reports information about that CI or if the data sources that provide data for a class are all equally trusted.

## Priority rule definitions

Priority rules for data sources can be defined for the entire class or for an individual attribute for a class. The two levels cannot be used at the same time on any single CI class.

- Priority rules can be changed from class level to attribute level or attribute level to class level for a particular CI class.
- Changing the priority rules or priority levels alters the definitions in the TADDM database. When the new rules are applied to the incoming data, there can be a delay before the system completely reflects the changed priority definitions for a given CI class.

### Class level priority rules

Class level priority rules provide an ordered list of data sources for the entire class. The priority of the data sources is determined by their position in the list. The first data source in the list has the highest

priority, the second data source in the list has the second highest priority, and so on. Data can be written to the database from a data source that is not defined in a priority rule. However, the data can be overwritten by any data source defined by a priority rule because it has the lowest priority for that class. Whenever the incoming data source has a higher priority, it updates the data in the database (either the entire class or a particular attribute). If the incoming data source has a lower priority than the data source that owns the data in the database, the incoming data is ignored, either for the class or for a particular attribute.

### **Attribute level priority rules**

Individual attribute priority rules behave the same way as class level rules. With one exception, the specific ordered list of data sources applies only for a particular attribute of the class. Each attribute in the class can have a different ordered list of data sources. The number of attributes in a class that can be prioritized using attribute level prioritization has a 192-character limitation. Therefore, the actual number depends on the attributes that are selected to be prioritized. The Discovery Management Console enforces the limit and informs you that an attempt was made to prioritize too many individual attributes.

### **Changing from attribute to class level priority rules**

If prioritization rules are changed from attribute to class level after data is persisted in the database, the detailed information about which data source provided which attribute value is not preserved the next time the data is loaded. Instead, one of the providing data sources for the existing data is selected as the owner for all of the data in the CI. This data source is used in the comparison with the incoming data source to determine if the data in the CI must be updated. This action occurs because class level prioritization requires that all data in a CI to come from a single provider.

### **No priority rules defined**

If no priority rules are defined, the latest data coming into the system updates the existing data.

### **Priority rules defined after data is in the database**

If priority rules are added after data is in the database, the existing data is given the lowest priority in the system. The existing data is overwritten by the incoming data, no matter what the incoming data sources priority. After data is written using the new rules, prioritization applies to future updates.

### **Priority rules are deleted after data in the database**

If priority rules are deleted after data is in the database, any incoming data updates the existing data.

## **Management software system (identifier)**

When data is saved into the TADDM database, a management software system (MSS) identifier is provided to define the identity of the data provider. The system automatically attempts to match the MSS identifier of a data provider with data source definitions that were defined for prioritization. If a match is made between an MSS identifier and a data source, then all priority rules that contain that data source are applied to the incoming data. The Details pane in the Discovery Management Console displays the management software system (MSS) that provides data on a particular CI.

## **Color codes for configuration items with priority rules**

From the **Attribute Prioritization** window, you can create a data source and prioritize rules for configuration items. The configuration items are color coded depending on what prioritization rules are associated with the attributes.

The configuration items in the **Attribute Prioritization** window are color coded to show if rules are applied to the configuration items.

The following colors are displayed in the window:

- In the Configuration Items pane, the CI class name is highlighted in blue if any of its attributes have priority rules assigned to it. If a CI class name is not highlighted, there are no rules defined for any of its attributes.
- In the TADDM Data Source List pane, the data source name is in green if it has an associated MSS identifier. The name is in blue if it has no associated MSS identifier. This situation occurs if the data source was not stored or if the data source entry is not defined correctly and has no matching MSS.

- In the upper right pane, in the Attribute Name and Source Object columns, the attribute line is highlighted according to the following schema:
  - Highlighted in blue if it has priority rules associated with it.
  - Highlighted in yellow if the source object it inherits the attribute from has priority rules associated with it.
  - Attribute is not highlighted if no priority rules are assigned to it and it does not inherit any rules.

Figure 1 on page 76 shows that the **adminState** attribute is highlighted in yellow. The CI named **Agent** has priority rules defined for the **adminState** attribute. The **TWSAgent** CI inherits its **adminState** attribute from that CI (Agent). The **accessMethod** attribute has no rules assigned to it and does not inherit any rules.

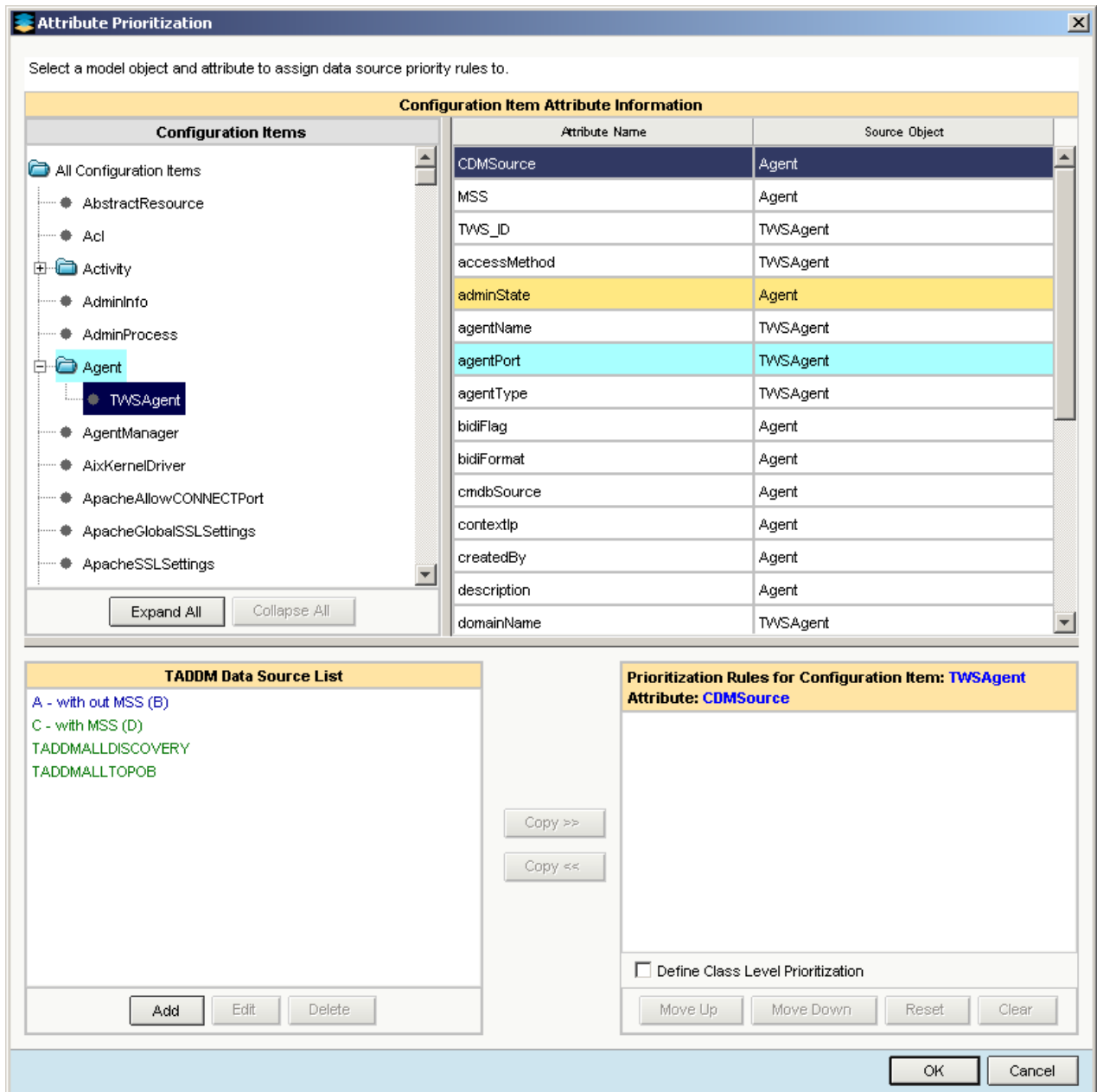


Figure 1. Attribute Prioritization window



## Adding prioritization rules to your configuration items (model objects)

You can use the Discovery Management Console to prioritize attributes for configuration items. Prioritization determines which data source takes priority over other data sources when updating configuration item (CI) attributes.

To prioritize the attributes for configuration items (CIs), complete the following steps:

1. From the Discovery Management Console, click **Edit > Prioritization Rules**.
2. In the TADDM Data Source List pane, click **Add**. The **Add Data Source** window is displayed.
3. Select one of the following data sources:

- **Create a Data Source for Discovery**

Select this option to create a source of data that is gathered from all sensors. The values cannot be changed and only one data source of this type can be created in the system. The name of the data source is TADDMALLDISCOVERY.

- **Create a Data Source for Topology**

Select this option to create a TADDM topology builder data source. The values cannot be changed. The name of the data source is TADDMALLTOPOB.

- **Create a Custom Data Source**

(Prior to **Fix Pack 4**) Enter values in the following fields:

- **Product Name**
- **Host Name**
- Optional: **Manufacturer Name**
- Optional: **Description**
- Optional: **MSS Assignment**

**Fix Pack 4** Enter values in the following fields:

- **Product Name**
- **Sub Component Name**
- **Sub Component Instance**
- Optional: **Description**

**Note:** This option is recommended over all other options as it covers all known use cases.

The 'Product Name', 'Sub Component Name' and 'Sub Component Instance' should be entered as displayed in management software system (MSS) with same capitalization, spelling, and punctuation. The MSS is displayed in **Edit>MSS** in Discovery Console.

- **Fix Pack 4 Create a MSS Data Source**

Select one of the following methods to enter the MSS Assignment:

- Type the product and host name entries exactly as displayed in the management software system (MSS). The name must have the same capitalization, spelling, and punctuation.
- Alternatively, from the **MSS Assignment (Optional)** list, select the appropriate MSS name. This list includes the MSS name that is associated with each sensor. All MSS names that you defined are displayed in this list. MSS names that result from loading data from IdML books or the bulk load program are also displayed.

**Note:** The **MSS Assignment** is optional. If you do not specify any values in the file, you can use a global setting from the etc/attrpriorot.properties file. The content of the file must comply with the following format:

```
<MSS>=<PRIORITY>
```

where <MSS> is the MSS name, and <PRIORITY> is the priority of the data source, 1 represents the highest priority. For example:

```
IBM:TADDm:ds.ibm.com:Discovery:WindowsComputerSystemSensor=10
```

4. Click **OK** to save the new data source.
5. In the **Configuration items** pane, select a model object. The attributes that are associated with this model object are displayed in the adjacent pane. Select an attribute name to assign to the new data source. The model object and attribute name, or names, that you selected are highlighted in blue at the top of the Prioritization Rules for Configuration Item and Attribute pane.
6. In the TADDM Data Source List pane, select the data source. Click **Copy**. The data source moves to the Prioritization Rules for Configuration Item and Attribute pane.
7. Optional: Select the **Define Class Level Prioritization** check box to apply this prioritization to all the attributes for the selected model object.
8. The Prioritization Rules for Configuration Item and Attribute pane, lists the data sources that are copied. To move the data source to the position that you want, select it and click **Move Up** or **Move Down**. The higher the position in this pane, the higher the priority for that data source.
9. Click **OK** to save your information.

## Data Management Portal

The Data Management Portal is the IBM Tivoli Application Dependency Discovery Manager (TADDM) web-based user interface for viewing and manipulating the data in a TADDM database. This user interface is applicable to a domain server deployment, to a synchronization server deployment, and to each storage server in a streaming server deployment. The user interface is very similar in all deployments, although in a synchronization server deployment, it has a few additional functions for adding and synchronizing domains.

## Discovery tasks

You can use the Data Management Portal to configure discoveries.

**Note:** Data Management Portal enables the configuration of discovery scopes and scope sets only. To perform operations on scope groups, refer to [“Scope pane” on page 114](#).

### Configuring a scope

You can use the Data Management Portal to configure a scope set and scope.

**Important:** Creating very large scopes can lead to performance issues, including a server crash.

To configure a scope set and scope, using the Data Management Portal, complete the following steps:

1. On the menu bar, click **Discovery > Scope**.  
The **Scope** pane is displayed.
2. To define a new discovery scope set, click **New scope set**.  
The **"New scope set"** window is displayed.
3. In the **Name** field, type the name for the new scope set.

**Important:** The scope set names cannot contain the following characters:

- ' (single quote)
- . (period)
- / (forward slash)

4. Click **OK**.

The new scope set is displayed in the **Scope sets** list.

5. To add the scope and contents to the scope set, select the scope set that you just created and click **New**.

The **"New scope"** window is displayed.

6. To add the settings for the scope, complete one of the following steps:

- In the Include panel, select **Subnet** from the list and do one of the following:
  - In the Address field, type the IP address of the subnet mask.
  - Move the slider to the IP address of the subnet mask.

The subnet mask entered must be a unique value within the scope set.

- In the Include panel, select **Range** from the list and in the Addresses field, type the first and the last IP addresses in the range. The values entered must be unique within the scope set.
- In the Include panel, select **Host** from the list and do one of the following:
  - In the **Addresses** field, type the IP address of the host.
  - In the **Description/hostname** field, type the host name.

The values entered must be unique within the scope set.

**Important:** If both the IP address and host name are defined and do not correspond to each other, the IP address takes precedence, and the host name is treated only as a description.

7. To exclude devices and hosts from your scope, in the Excludes panel, click **Add** and complete one of the following steps:

- Select **Subnet** from the list and do one of the following:
  - a. In the Address field, type the IP address of the subnet mask.
  - b. Move the slider to the IP address of the subnet mask.
- Select **Range** from the list, and type the first and last IP addresses in the range.
- Select **Host** from the list, and type the IP address of the host name.

8. To save the scope, click **OK**.

The new scope is displayed in the list.

## Changing a scope

You can use the Data Management Portal to change an existing discovery scope.

**Important:** Creating very large scopes can lead to performance issues, including a server crash.

To change an existing discovery scope, using the Data Management Portal, complete the following steps:

1. On the menu bar, click **Discovery > Scope**.

The **Scope** pane is displayed.

2. From the **Scope sets** list, select a scope set.

The list of scopes for that scope set are listed to the right.

3. From the list of scopes, select a scope and click **Edit**.

The **"Edit scope"** window is displayed.

4. To change the settings for the scope, complete one of the following steps:

- To change a subnet, in the **Address** field, type the IP address of the subnet. The subnet value entered must be a unique value within the scope set. Continue to Step 5.
- To change a range of devices, in the **Address** field, type the first and the last IP addresses in the range. The values entered must be unique within the scope set. Continue to Step 5.
- To change a specific device, in the **Address** field, type the IP address, or in the **Description/hostname** field, type the fully-qualified host name. The values entered must be unique within the scope set. Continue to Step 6.

5. To exclude devices and hosts from the discovery scope, in the Excludes panel, click **Add** and complete one of the following steps:

- Select **Host** from the list and type the IP address of the host name.

- Select **Subnet** from the list and type the IP address of the subnet.
  - Select **Range** from the list and type the first and last IP addresses in the range.
6. To save the scope, click **OK**. The new changes are applied to the scope.

## Deleting a scope

You can use the Data Management Portal to delete a scope.

To delete a scope, using the Data Management Portal, complete the following steps:

1. On the menu bar, click **Discovery > Scope**.  
The **Scope** pane is displayed.
2. From the **Scope sets** list, select the scope set that contains the scope that you want to delete.  
The list of scopes for that scope set are listed to the right.
3. From the list of scopes, select a scope and click **Delete**.  
A message window is displayed.
4. To delete the scope, click **Yes**.  
The scope is deleted from the scope set.

## Deleting a scope set

You can use the Data Management Portal to delete a scope set.

To delete a scope set, using the Data Management Portal, complete the following steps:

1. On the menu bar, click **Discovery > Scope**.  
The **Scope** pane is displayed.
2. From the **Scope sets** list, select the scope set that you want to delete, and click **Delete scope set**.  
A message window is displayed.
3. To delete the scope set, click **Yes**.  
The scope set is deleted.

## Adding custom servers

You can use the Data Management Portal to add custom servers.

To add a custom server, complete the following steps from the Data Management Portal:

1. In the Functions pane, click **Discovery > Custom Servers**.  
The **Custom Servers** pane is displayed.
2. In the **Custom Servers** pane, click **New**.  
The **General Info & Criteria** tab of the **Custom Server Details** notebook is displayed.
3. To enable the custom server definition, click **Enabled**.
4. In the **Name** field, type the name of the custom server.
5. From the **Type** list, select the type of custom server that you are adding.
6. To set the action, complete one of the following steps:
  - Click **Discover** if you want to discover all instances of the server.
  - Click **Ignore** if you want to suppress discovery of all instances of the server.
7. To select an icon to associate with the custom server, click **Browse** and select the icon that you want to use.
8. In the Identifying Criteria panel, complete one of the following steps:
  - To match all of the identifying criteria, click **All of the following conditions match (logical AND)**.
  - To match any of the identifying criteria, click **Any of the following conditions match (logical OR)**.
9. Complete the following steps to define the criteria for the custom server:

- a) From the first list, select the criterion type.
  - b) From the second list, select the operator.
  - c) In the field provided, type the text argument for the criterion type and operator.
10. To add the new criterion, click **+**.
  11. To add configuration files, click the **Config Files** tab.  
The **Config Files** page is displayed.
  12. On the **Config Files** page, click **New**.  
The **Search Path for Capture File** window is displayed.
  13. From the **Type** list, select one of the file types to capture:
    - Config File
    - Software Module
    - Application Descriptor Directory/File
  14. From the **Search Path** list, select one of the following search paths for the configuration file:
    - /**  
The root of the file system.
    - \$PWD**  
The current working directory of the running program.
    - \$HOME**  
The home directory of the user ID of the running program.
    - C:**  
A directory on your local computer.
    - %ProgramFiles%**  
The program files directory.
    - %SystemRoot%**  
The system root directory.

Type the path and file name of the configuration file in the text box, or type \* (asterisk) to specify all files in the selected directory.
  15. To capture the contents of the configuration file, click **Capture file contents** and optionally specify the maximum number of bytes of the captured configuration file.
  16. To recurse through the directory structure to search for the specified file, select **Recurse Directory Search** (if you use TADDM 7.3.0.3, or later), or **Recurse Directory Content** (if you use TADDM 7.3.0.2, or earlier).
  17. To save the settings for your custom server, click **OK**.

## Editing a custom server

You can use the Data Management Portal to edit a custom server.

To edit a custom server, complete the following steps using the Data Management Portal:

1. In the Functions pane, click **Discovery > Custom Servers**.  
The **Custom Servers** pane is displayed.
2. In the **Custom Servers** pane, click **Edit**.  
The **Custom Server Details** notebook is displayed, with the **Name** and **Type** fields disabled. These fields cannot be changed.
3. To change the other fields in the **Custom Server Details** notebook, see [“Adding custom servers” on page 80](#).
4. To refresh the information about the custom server you just changed, run another discovery.  
To improve the speed of the discovery process, limit the active scope of the discovery to the new component.

## Copying a custom server

You can create a custom server based on an existing one. This is done by copying a server listed in the **Custom Servers** pane and issuing it a unique name.

To copy a custom server, complete the following steps from the Data Management Portal:

1. In the Functions pane, click **Discovery > Custom Servers**.  
The **Custom Servers** pane is displayed.
2. In the **Custom Servers** pane, select the custom server that you want to copy and click **Copy**.  
The **Custom Server Details** window is displayed.
3. In the **Name** field, type the name of the new custom server.
4. If appropriate, type new values for one or more properties of the new custom server.
5. To save the new custom server, click **OK**.

## Deleting a custom server

You can use the Data Management Portal to delete a custom server.

To delete a custom server, complete the following steps from the Data Management Portal:

1. In the Functions pane, click **Discovery > Custom Servers**.  
The **Custom Servers** pane is displayed.
2. In the **Custom Servers** pane, select the custom server that you want to delete and click **Delete**.  
A message window is displayed.
3. To delete the custom server, click **Yes** in the message window.
4. To confirm the deletion, ensure that the custom server is not listed in the **Custom Servers** pane.

## Working with grouping patterns

In the **Grouping Patterns** pane, in the Data Management Portal, you can create, display and edit grouping patterns. Grouping patterns can be used to have your CIs automatically grouped into custom collections of type business application, collection, or access collection.

For details, see [“Creating business applications with grouping patterns” on page 179](#).

See also the *Managing grouping patterns* and *Maintaining Grouping Patterns using REST API* topics in the *TADDM SDK Developer's Guide*.

## Viewing the discovery history

Each time that a discovery is run, the Data Management Portal updates the discovery activity and error information that is displayed in the **History** pane.

You can view the discovery history, including the associated activity and error information, in the **History** pane. By default, information about the last ten discoveries is displayed.

It might take a long time to retrieve and display the discovery history in the History pane. As an alternative, consider using the Sensor Done Events by Run BIRT report.

To view a discovery history, complete the following steps from the Data Management Portal:

1. In the Functions pane, click **Discovery > History**.  
The **History** pane is displayed.
2. To display information about a discovery, select an entry in the table.  
A second table of data is displayed. This table provides a list of sensors and the host name, IP address, date, status, and description for each sensor.
3. To display the scopes that are included in the discovery, click **Scope Details**. The **Scope List** window is displayed.  
To close the **Scope List** window, click **Close**.

4. To display information about an event, select the event and click **Event Details**. The **Discovery Event** window is displayed.

To close the **Discovery Event** window, click **Close**.

## Identifying unknown servers

You can identify unknown servers running on a computer system, and use this information to create a custom server template that you can use for future discoveries.

Unknown servers are identified after a discovery by a topology build agent. The topology build agent runs in the background on a periodic basis, depending on the value of the configured frequency, so unknown servers might not be recognized immediately after a discovery completes. Every four hours is the default frequency at which the topology build agent runs.

To set the frequency of the background agent, configure the following property in the `collation.properties` file:

```
com.ibm.cdb.topobuilder.groupinterval.background=frequency
```

where *frequency* is the frequency, in hours, of the background agent. The default value is 4.0.

To identify unknown servers, complete the following steps in the Data Management Portal:

1. In the **Discovered Components** pane, select one or more computer systems that you want to check for unknown servers.
2. Click **Actions > Unknown Servers**.  
The **Unknown Servers** pane is displayed.
3. To create a custom server based on an unknown server, select an unknown server and click **Create Custom Server**.

The unknown server will remain marked as unknown until after a custom server template has been created based on the unknown server, and a discovery performed.

## Manually merging discovered configuration items

Manual merging is the process where you combine two or more configuration item objects (CI) into one CI. You can use this process to remove duplicate CIs.

When merging CIs, a single CI is selected from the list of CIs to be merged. This CI is called the "durable" CI and is retained at the end of the merge operation. The other CIs are called "transient" CIs and are deleted at the end of the merge operation.

The following rules apply to manually merging CIs:

- Only CIs of the same type can be merged.
- When CIs are merged, only the attributes of primitive types (for example, string and integer) are transferred from the transient CI to the durable CI. This transfer occurs only if the durable CI does not already have a value for that attribute. Arrays and objects associated with the transient CI are not transferred.
- When a transient CI is deleted, all of its related CIs are also deleted. For example, if a ComputerSystem CI is deleted then the operating system CI running on the computer system and all the software installations on the operating system are deleted.
- If a CI that is designated as a transient object and is later rediscovered or reloaded through the bulkload facility, it updates the durable object. This method does not result in a second instance of the CI.
- Merging is not currently supported for Business Systems or Business Applications.



**Warning:** Do not display / navigate a transient CI when the merge process is ongoing. This can cause the error.log file to trace a null point exceptions, and the merge to fail without displaying an error message.

To merge CIs, complete the following steps in the Data Management Portal:

1. Do one of the following:

- In the **Discovered Components** pane, select the CIs that you want to merge. Click **Actions > Merge**.
- In a topology view, select the CIs that you want to merge, right-click them and select **Merge**.

The **Merge Component** window is displayed.

2. In the **Display Name** list, select the CI to retain at the end of the merge (durable CI.) Click **Mark as durable**.

3. The remaining CIs are merged in the order that is displayed in the **Display Name** list. To change the priority, select a CI and click **Move Up** or **Move Down** to change the order.

4. Click **OK** to save your information.

Information concerning merge operations are recorded in the `$COLLATION_HOME/log/services/ReconciliationMerge.log` file.

If two CIs are mistakenly merged together attempting to rediscover or reload, the transient object results in updating the durable object and does not re-create the original transient CI. The durable CI must be deleted and the durable and transient CIs rediscovered or reloaded.

## Manually deleting merged configuration items

If you no longer want manually merged configuration items, then you can manually delete them and discover the original configuration items again. If two CIs are mistakenly merged together or duplicate child objects are displayed you can manually delete them.

To manually delete merged CIs, complete the following steps from the Data Management Portal:

1. In the Discovered Components pane, select the CI (durable object) that you want to delete.
2. Click **Actions > Delete**. The **Delete Items** window is displayed.
3. In the **Delete Items** window, select the CI and click **OK**. If the CI is used by another component, a confirmation window is displayed.
4. Click **OK** to delete the CI.

Perform a discovery to see the original CIs.

## Creating components

You can create a component using the **Create Component** wizard.

The specific pages displayed in the **Create Component** wizard depend on the type of component you are creating.

To create a component, complete the following steps:

1. On the menu bar, click **Edit > Create Component**.

The General Information page of the **Create Component** wizard is displayed.

2. In the **Name** field, type the component name.
3. From the **Type** list, select the type of the component you want to create.
4. Click **Next**.

The next page of the **Create Component** wizard is displayed.

5. Depending on the type of component you want to create, different pages of the **Create Component** wizard are displayed.

Complete any of the following tasks that are appropriate:

- In the Server Information page of the **Create Component** wizard, complete the following tasks:
  - a. In the **Available** list, select the computers you want to add.
  - b. Click **Add**.
- In the IP Information page of the **Create Component** wizard, complete the following tasks:



- a. In the **Host name** field, type the host name of the computer you want to add.
  - b. In the **IP address** field, type the IP address of the computer you want to add. If appropriate, move the slider to the specify the subnet mask.
  - c. Click **Add**.
6. Click **Next**.

The Administrative Information page of the **Create Component** wizard is displayed.
7. Optional: Specify some or all of the following information:
  - Admin contact
  - Escalation contact
  - Tracking number
  - Site
  - Group name
  - Notes
8. If one or more extended attributes have been defined for the type of component you are creating, click **Next**.

The Extended Attributes page of the **Create Component** wizard is displayed.
9. Optional: In the Extended Attributes page, specify a value for one or more of the extended attributes listed.
10. Click **Finish**.

## Editing components

You can edit an existing component.

The specific pages displayed in the **Edit Component** notebook depend on the type of component you are editing.

To edit a component, complete the following steps:

1. In the **Discovered Components** pane, select the component you want to edit.
2. Click **Actions > Edit**.

The **Edit Component** notebook is displayed.
3. Click the tab containing the information you want to edit.

Depending on the type of component you are editing, some of the following tabs are available:

  - General Information
  - Server Information
  - IP Information
  - Administrative Information
  - Extended Attributes
4. Update the component information.
5. Click **OK**.

## Topology tasks

You can use the Data Management Portal to view graphical topology information.

### Displaying an overview topology

An overview topology contains all the configuration items in a category. An overview topology can be viewed in the Data Management Portal using the Topology function.

To display an overview topology, complete the following steps in the Data Management Portal:

1. In the Functions pane, click **Topology**.
2. Click **Business Applications**.

An overview topology for the item is displayed.

When you perform a new discovery or make changes on the synchronization server, such as adding a business application or service, the synchronization server does not automatically reflect the changes in the Data Management Portal. You can view the latest changes by reloading the view.

## Displaying a specialized topology

You can view a specialized topology for configuration items (CI) of certain types in the Data Management Portal.

Specialized topologies are available for a number of CI types. You can launch these topologies from the Discovered Components pane. [Table 15 on page 86](#) lists the specialized topologies and the CI types for which they are available.

<i>Table 15. Specialized topologies</i>	
<b>Topology name</b>	<b>CI type</b>
AIX Topology	AIX computer system
BladeCenter Topology	computer system of subtype BladeCenter
Business App Physical Topology	business application, vApp, SAP System, Siebel Enterprise
Business App Software Topology	business application, vApp, SAP System, Siebel Enterprise
Citrix Farm Topology	Citrix Farm
CSM Cluster Topology	Configuration Management cluster
CSM Cluster L2 Topology	Configuration Management cluster
CSM Cluster system dep. Topology	Configuration Management cluster
Exchange 2007 Topology	Exchange 2007 group
HACMP Cluster Topology	HACMP cluster
HMC and LPAR Topology	System p computer system
HyperV Virtual System Topology	Windows computer system
MQ Cluster Topology	WebSphere MQ cluster
MSCluster Topology	Microsoft cluster
Oracle ASM Topology	Oracle Automatic Storage Management (ASM)
Oracle RAC Topology	Oracle Real Application Clusters (RAC)
Pix Firewall Topology	Cisco Pix computer system
Physical Topology	Collection
Relationship Topology	Collection
Storage sub system Physical Topology	StorageSubSystem
Subnet drilldown	IP subnet, IPv4 network, IPv6 network
SunFireComputerSystem Topology	Sun Fire computer system
Switch-Apps Topology	computer system of subtype router, bridge, or switch
Switch-IP Devices Topology	computer system of subtype router, bridge, or switch

Table 15. Specialized topologies (continued)	
Topology name	CI type
SysImager Cluster L2 Topology	Configuration Management cluster
SysImager Cluster system dep. Topology	Configuration Management cluster
SysImager Cluster Topology	Configuration Management cluster
System Connection Topology	business application, vApp, SAP System, Siebel Enterprise
VERITAS Cluster Topology	Veritas cluster
VIOS Storage Topology	System p computer system
Virtual Center Topology	VMware VirtualCenter
Virtual Center VirtualSwitch Topology	VMware VirtualCenter
Virtual Systems Topology	VMware unitary computer system
VirtualSwitch Topology	VMware virtual switch
WebSphere Cell Topology	WebSphere Cell
WebSphere XS Cache Topology	WebSphere XS Cache
WebSphere XS Cache Node Topology	WebSphere XS Node
Z Topology	zSeries
ZOS Topology	z/OS computer system


To display a specialized topology, complete the following steps in the Data Management Portal:

1. In the Discovered Components pane, navigate to and select the CI for which you want to view a topology.
2. Click **Actions** and select **Show topology\_name**. If a *topology\_name* option is not available, select **Show Topology**.  
A topology for the CI is displayed.

## Exporting a topology

You can export the currently displayed topology to an image file.

To export a topology, complete the following steps in the Data Management Portal:

1. In the Functions pane, click **Topology**.
2. Click **Business Applications**.  
An overview topology for the item is displayed.
3. Click the  icon.  
The **Export Topology** window is displayed.
4. Select the type of file to which you want to export the topology.  
The following options are available:
  - JPEG format (file type is JPG)
  - Portable Network Graphics (file type is PNG)
  - Scalable Vector Graphics (file type is SVG)
5. If prompted to do so, for the JPG and PNG file types, specify the image resolution by typing the image height and width.  
For some topologies, larger resolutions might be needed for clarity. The maximum resolution is 9999 x 9999 pixels. For the SVG file type, the image height and width are irrelevant.

**Note:** **Fix Pack 3** When you export large business application topologies to the SVG file type, you can use a dedicated command line API, the `bizappscli` tool, which generates smaller files and allows you to compress the file to the `.zip` format to even further reduce the file size. For details, see [“Actions for exporting topologies to the SVG format” on page 230.](#)

6. Click **Export**.

Depending on the configuration of your browser, you are prompted to save the image file, or the image is displayed in the browser.

## Manually defining dependencies between configuration items

You can define dependencies between configuration items (CIs) by creating an XML definition file. You define an SQL select query within the definition file to select the dependencies. The dependencies are automatically created and are displayed in the topology and Details pane in the Data Management Portal.

1. On the TADDM server, create a *definition\_filename.xml* file in the `$COLLATION_HOME/etc/dependencies` directory. For enterprise deployments, the custom definition files must be stored in the primary storage server.
2. Edit the *definition\_filename.xml* file. This file must contain the following attributes:

**Label**

A short definition name.

**Type**

A dependency type such as a dependency class name. For example, *app.dependencies.ServiceDependency*.

**Description (optional)**

A brief description of the dependencies.

**Query**

An SQL query that must return at least two columns containing source and target aliases. The aliases contain a pair of globally unique identifiers (GUIDs) that are used to create a dependency between the CIs.

The XML definition file must be compliant with the XML schema definition (XSD) file. The *schema.xsd* file is located in the `$COLLATION_HOME/etc/dependencies` directory.

The topology builder builds the relations and dependencies between the discovered items. The topology builder runs a list of agents at specified intervals. After the agent runs the query, TADDM builds the custom defined dependencies.

When the dependencies are no longer valid they are automatically deleted. Removing the *definition\_filename.xml* file, leaves the already created dependencies intact but changing the query can lead to the removal of created dependencies.

To remove unwanted custom dependencies create an empty query as shown in the following example.

The *example.xml* file is in the `$COLLATION_HOME/etc/dependencies` directory and shows the steps to define a custom dependency definition file.

```
<dependency xsi:noNamespaceSchemaLocation="schema.xsd">
  <label>Example</label>
  <type>app.dependencies.ApplicationToApplicationDependency</type>
  <query>SELECT guid_x AS SOURCE, guid_x AS TARGET FROM appsvr WHERE 0 = 1</query>
</dependency>
```

## Analytics tasks

You can use the Data Management Portal to perform analytics and generate reports.

### Displaying component comparison information

You can create a Component Comparisons report in the Data Management Portal.

You can compare two or more components of the same type and create a report based on the comparison. Two modes of comparison, basic and deep, are available, which will compare elements of the components in a basic or in an in-depth, drilled down fashion. The type of comparison made is determined by the components you have selected.

When comparing custom collections, you can compare only two at a time. In basic mode, crucial elements of the collection are compared, such as Custom Collection attributes (like displayName, hierarchyType, extended Attributes), CoreCIs displayName (grouped by tierName and Type), Number of Nodes (grouped by tierName and Type), and basic GroupingPattern information (Name and Extended Attributes). In deep mode, in addition to comparing the basic mode elements, comparison is made of the displayName for all Nodes (grouped by TierName and Type).

To display a Component Comparison report, complete the following steps from the Data Management Portal:

1. In the Functions pane, click **Analytics > Component Comparison** to define the parameters for a component comparison report.

The Component Comparison tab of the **Component Comparison** pane is displayed.

2. In the Components section, complete the following steps:

- a) From the **Version** list, select the discovery version against which you want to perform the component comparison.

- b) From the **Available** list, select the components you want to compare.

- c) Click **Add**.

- d) Optional: To set the component as the key to compare all other components against, select an included component, and click **Set as Key**.

You can set a component as a key in cases when you have a known component that has a correct configuration.

This component acts as the master against which all other included components are compared. When a component with a different configuration than the key is encountered, the application highlights the component and the differing configuration in the color red. In cases when you do not pick a key, the first component is automatically assigned as the key. This can happen when you do not have a known good component to compare against. For example, you might have a cluster that is experiencing problems, but you do not know which servers are good or bad.

3. Select the options for the component comparison.

- a) For Level, select either **Deep** or **Basic**.

- b) For Include Infrastructure Services, select either **Yes** or **No**.

- c) For the Available Components, select either **Yes** or **No**.

4. To run the report, click **Run Report**.

The report is displayed in the **Results** tab of the **Component Comparison** pane.

5. Click the **Results** tab to view the **Component Comparison: Results** pane.

### Displaying a change history report

You can display a change history report for all of your discovered configuration items (CIs) in the Data Management Portal using the Analytics function.

**Note:** The label attribute of a CI is not set when the CI is discovered. It is set for the first time when you change the displayed name of the CI from the name with which it was created. In the change history report, the field displaying the old label value is blank after you change the CI label for the first time.

You can exclude particular elements from the change history report by configuring the following file:

```
$COLLATION_HOME/etc/changeserver.xml
```

In general, including an element in `changeserver.xml` results in it being excluded from the change history, but consider the following configuration rules:

- If you add a class name with no attributes, then that class, its specializing classes, and all the classes that it contains are ignored.
- If you add a class with attributes, then the attributes and the specializing classes of that class are ignored.
- If you add a package name, then that entire package and all child packages are ignored.
- If you add an attribute name, then that attribute is ignored across all classes.

When configuring the objects and attributes to be ignored, specify API names and object names with an uppercase initial letter, for example `WindowsService`, and attribute names with a lowercase initial letter, for example `processId`.

After you make changes to the `changeserver.xml` file, you must restart the TADDM server for the changes to take effect.

To display change history information, complete the following steps:

1. In the Functions pane, click **Analytics**.
2. Click the **Change History** item.  
The **Change History** pane is displayed.
3. Specify a time frame for the report by completing the **Timeframe** section.  
You can specify an absolute or a relative time range. If the start time and the end time are the same, no change history information is displayed.
4. In the **Components** panel, from the **Component Type** list, select the type of component about which you want to create a change history report.  
In the **Available Components** list, all components of the selected type are displayed.
5. In the **Available Components** list, click each component you want to include in the report, and click **Add**.  
The added components are displayed in the **Included Components** list.
6. Click **Run Report**.  
The report results are displayed in the **Results** tab.
7. If both the old value and the new value are text values and are more than 100 characters in length, they are truncated in the report. To see the complete values, click **Show Details**.
8. To compare old and new text values that are more than 100 characters in length, click **Show Differences**.  
The lines that differ between the two values are displayed. For each difference, the following information is displayed:
  - line number
  - type of change (added, removed, or changed)
  - new value
  - old valueTo save the changes to a file, click **Save**. To close the window, click **Close**.
9. To sort the report by attribute, click the column heading for that attribute. For example, you can sort the report by date by clicking the **Date** column heading. Clicking inside a heading changes the sort order for the attribute between ascending and descending order. You can also change column widths by clicking the heading row at any column border and dragging it to the left or right.
10. To export the contents of the table of results to a file, click **Export**.  
The **"Export Report as"** window is displayed.

11. From the **Save As** list, select the file type to which you want to export the information in the table of results.

The available file formats are:

- PDF
- CSV
- XML

**Note:** If you are using the Microsoft Internet Explorer browser and connecting using a secure session, you cannot export the report information to a file. The following alternatives are available:

- Use an alternative web browser.
- Use Tivoli Common Reporting for viewing and administering reports.

12. To export the report, click **Export**.

Your browser will open the exported file based on the file-handling preference specified for that format of file.

## Working with custom queries

You can create queries for reports that are based on discovered data that is stored in the TADDM database.

In the **Custom Query** pane of the Data Management Portal, you can formulate a simple query by selecting components and specifying criteria. This SQL-like query returns views from a single table for the component specified in the TADDM database. This data can be in a different format than what is displayed in the **Details** pane, because it shows the raw database values, not the formatted values that are displayed in the **Details** pane.

The **Saved Queries** tab of the **Custom Query** pane displays existing custom queries.

**Note:** The data required for the **Component type** list on the Edit Query tab must be loaded before this option is available. This process takes place in the background beginning when you log on to the Data Management Portal interface and can take several minutes to complete. If you try to access the **Component type** list immediately after logging on, you might need to wait for the loading to complete.

### *Creating a custom query*

You can create a custom query, or you can make a copy of an existing query, change the properties, as necessary, and save it as a new custom query.

When creating a custom query, you must specify a component type. The **Component Type** lists objects found by TADDM discoveries. If no TADDM discoveries have taken place, no component information is available, and so, you cannot create a custom query.

When you select a component type, the default attributes for that component type are displayed. You can change the set of default attributes for a component.

To create a custom query, complete the following steps in the Data Management Portal:

1. In the Functions pane, click **Analytics > Custom Query**.

The **Custom Query** pane is displayed.

2. Do one of the following:

- Click **New**. The **New Query** tab is displayed.
- In the **Saved Queries** tab, select the custom query you want to copy and click **Copy**. The **Edit Query** tab is displayed.

3. In the **Name** field, type the name of the custom query.

4. From the **Component Type** list, select a component type.

**Note:** If you are copying an existing custom query, you cannot change the component type of the new custom query.

The default attributes for that component type are displayed.

5. Select **Match all criteria** or **Match any criteria** to specify a logical AND or logical OR for multiple comparison criteria.
6. Specify criteria to match one or more of the displayed attributes. The displayed attributes depend on type of component for which you are creating a custom query. For each criterion you want to specify, complete the following steps:
  - a) Select a criterion type. The criterion types available depend on the attribute type.
  - b) In the attribute value field, specify a criterion value. As you type an attribute value for a string attribute, a list of suggestions might be presented. If you select a suggestion from the list, that value is placed into the attribute value field.

Attribute value suggestions are displayed where the inputted text forms any part of the name of a configuration item discovered by TADDM. This field is not case-sensitive.
- Note:** The "not equals" and "!=" criteria do not match null values. They match any non-null attribute value that does not equal the provided one. The attribute value cannot contain a " " (quotation mark) or ' ' (apostrophe) character, for example nc "10.
7. To run the custom query before saving it, click **Run Query**.

Running the query gives you the opportunity to examine the results and ensure that the query performs as expected before saving it.
8. To save the custom query, click **Save**.

### ***Configuring custom query attributes***

You can configure the attributes used in a custom query.

To configure custom query attributes, complete the following steps in the Data Management Portal:

1. In the Functions pane, click **Analytics > Custom Query**.

The **Custom Query** pane is displayed.
2. Do one of the following:
  - Click **New**. The **New Query** tab is displayed. From the **Component Type** list, select the component for which you want to create a custom query.
  - In the **Saved Queries** tab, select the custom query you want to copy, and click **Copy**. The **Edit Query** tab is displayed.
  - In the **Saved Queries** tab, select the custom query you want to edit, and click **Edit**. The **Edit Query** tab is displayed.
3. Click **Configure**.

The **Configure Attributes** window is displayed. By default, the **Available Attributes** list displays all available attributes for the selected component. To filter the Attributes displayed, in the **Filter** field, type an attribute name, or part of an attribute name. Only attributes matching the entered text are displayed in the **Available Attributes** list.
4. To add an attribute to the custom query, do one of the following:
  - In the **Available Attributes** list, select the attribute you want to add, and click **Add**.
  - In the **Available Attributes** list, double-click the attribute you want to add.

The attribute is displayed in the **Included Attributes** list.
5. To remove an attribute from the custom query, do one of the following:
  - In the **Included Attributes** list, select the attribute you want to remove, and click **Remove**.
  - In the **Included Attributes** list, double-click the attribute you want to remove.

The attribute is removed from the **Included Attributes** list.
6. To restore the **Included Attributes** list with the default attributes for the current component type, click **Restore Defaults**.



7. To save the attributes currently in the **Included Attributes** list as the default list of attributes for the current component type, click **Save as Defaults**.
8. Click **OK**.

### ***Running a custom query***

You can run an existing custom query, view the results in a table, and view additional information about each of the objects listed in the query results.

To run a custom query and view the results, complete the following steps in the Data Management Portal:

1. In the Functions pane, click **Analytics > Custom Query**.

The **Custom Query** pane is displayed.

2. Do one of the following:

- From the **Saved Queries** tab, select a query and click **Run Query**.
- From the **Edit Query** tab, click **Run Query**.
- In the **Edit Query** tab, ensure that from the **Component Type** list, a component has been selected. Click **Run Query**.

**Note:** You do not need to specify a name to run a custom query, but you must specify a name to save a custom query.

The **Results** tab is displayed. The columns displayed in the table correspond to the attributes selected in the query.

3. To view detailed information about an object listed in the query results, in the **Results** pane, select a query result and click **Details**.

The **Details** notebook is displayed.

4. To view the relationships of an object listed in the query results, in the **Results** pane, select a query result and click **Explore**.

A node centered topology graph of the object is displayed.

5. To add an object listed in the query results to the list of components to be compared, select a query result and click **Mark For Comparison**.

The **Component Comparison** window is displayed containing the selected component. To continue with the component comparison, complete the following steps:

- a. In the **Custom Query** pane, click the second component to be compared.

**Important:** You do not need to close the **Component Comparison** window prior to clicking on another component in the **Custom Query** pane.

The **Component Comparison** window is displayed containing the second selected component.

- b. Repeat this step for as many additional components that you want to compare.
- c. In the **Component Comparison** window, select the components that you want to compare and click **Compare**. The **Component Comparison** pane is displayed.

For more information about component comparison, see [“Displaying component comparison information”](#) on page 89 and [“Component Comparison: Results pane”](#) on page 147.

6. To view the change history of an object listed in the query results, in the **Results** pane, select a query result and click **Changes**.

The change history of the object is displayed.

7. To export the query results to a file (for example, Adobe Portable Document Format, Comma Separated Values, or XML), complete the following steps:

- a) In the **Results** pane, select a query result and click **Save**.

The **Export** pane, is displayed.

- b) From the **Format** list, select the format in which you want to save the query results.
- c) Click **Save**.

**Note:** If you are using the Microsoft Internet Explorer browser and connecting using a secure session, you cannot export the report information to a file. The following alternatives are available:

- Use an alternative web browser.
- Use Tivoli Common Reporting for viewing and administering reports.

### ***Editing a custom query***

You can edit the properties of a custom query.

To edit a custom query, complete the following steps in the Data Management Portal:

1. In the Functions pane, click **Analytics > Custom Query**.

The **Custom Query** pane is displayed.

2. In the **Saved Queries** tab, select the custom query you want to edit.

3. Click **Edit**.

The **Edit Query** tab is displayed.

4. Set new values for some or all of the attributes displayed.

**Note:** You cannot change the component type of the query.

5. To run the custom query before saving it, click **Run Query**.

Running the query at this point gives you the opportunity to examine the results and ensure that the query performs as expected before saving it.

6. To save the edited query, click **Save**.

### ***Deleting a custom query***

You can delete a custom query from the table of saved queries.

To delete a custom query from the list of saved queries, complete the following steps in the Data Management Portal:

1. In the Functions pane, click **Analytics > Custom Query**.

The **Custom Query** pane is displayed.

2. In the **Saved Queries** tab, select the custom query you want to delete.

3. Click **Delete**.

The custom query is removed from the table of saved queries.

## **Displaying inventory summary information**

You can display an inventory summary in the Data Management Portal using the Analytics function.

To display inventory summary information, complete the following steps:

1. In the Functions pane, click **Analytics**.

2. Click the **Inventory Summary** item.

The **Inventory Summary** pane opens.

3. Click an entity in the **Component Type** or **Inventory Detail** column.

The **Inventory Details** pane opens.

4. In the **Inventory Details** pane, select a component and click one of the following buttons depending on the type of information that you want to display:

- Click **Details** to display the **Details** pane for the selected component.
- Click **Changes** to display the **Change History** pane for the selected component.
- Click **Save** to save the Inventory Summary report for the selected component.
- Click **Mark For Comparison** to add this component to the list of components to be compared. The **Component Comparison** window is displayed containing the selected component. To continue with the component comparison, complete the following steps:
  - a. In the **Inventory Details** pane, click the second component to be compared.

**Important:** You do not need to close the **Component Comparison** window prior to clicking on another component in the **Inventory Details** pane.

The **Component Comparison** window is displayed containing the second selected component.

- b. Repeat this step for as many additional components that you want to compare.
- c. In the **Component Comparison** window, select the components that you want to compare and click **Compare**. The **Component Comparison** pane is displayed.

**Important:** For more information about component comparison, see [“Displaying component comparison information” on page 89](#) and [“Component Comparison pane” on page 146](#).

**Note:** With data-level security enabled, when you have restricted access to CIs and you run the inventory summary report, you can see a complete list of CIs as if you had administrative rights. But you cannot see any details about the restricted CIs. The "Access denied" message is displayed when you try to display, for example, the **Details** pane or the **Change History** pane.

## Displaying change history information using the Inventory Summary

You can display change history for a selected configuration item in the Data Management Portal using the Inventory Summary function.

To display change history information, complete the following steps:

1. In the Functions pane, click **Analytics**.
2. Click the **Inventory Summary** item.  
The **Inventory Summary** pane is displayed.
3. In the **Inventory Summary** pane, click an entity in the **Component Type** or **Inventory Detail** column.  
The **Inventory Details** pane is displayed.
4. In the **Inventory Details** pane, click an entity in the **Display Name** column and click **Changes**.  
The **Change History** pane is displayed.
5. To display change history information for a specific timeframe, specify the time range for the report by completing the **Timeframe** section. You can specify an absolute or relative time range. Enter the start and end dates and times for the period that you want in the report.
6. Click **Show Changes** to display change history information for the specified timeframe.

## Displaying application summary information

You can display your application summary details information and history in the Data Management Portal using the **Analytics** function.

To display a summary of your business applications, complete the following steps:

1. In the Functions pane, click **Analytics**.
2. Click the **Application Summary** item.  
The **Application Summary** pane opens.
3. In the **Application Summary** pane, select an entity in the **Application Name** column and do one of the following steps:
  - Click **Details** to display the **Application Summary Details** pane.
  - Click **Change** to display the **Change History** pane.
  - Click **Software Topology** to display the topology of software for the business application.
  - Click **Physical Topology** to display the topology of hardware used by the business application.
  - Click **Inventory** to display the **Inventory Summary** pane.
  - Click **Mark For Comparison** to add this component to the list of components to be compared.

The **Component Comparison** window is displayed containing the selected component. To continue with the component comparison, complete the following steps:

- a. In the **Application Summary** pane, click the second component to be compared.

**Important:** You do not need to close the **Component Comparison** window prior to clicking on another component in the **Application Summary** pane.

The **Component Comparison** window is displayed containing the second selected component.

- b. Repeat this step for as many additional components that you want to compare.
- c. In the **Component Comparison** window, select the components that you want to compare and click **Compare**. The **Component Comparison** pane is displayed.

For more information about component comparison, see [“Displaying component comparison information” on page 89](#) and [“Component Comparison pane” on page 146](#).

## Displaying system inventory information

You can display a system inventory report for all of your discovered configuration items (CIs) in the Data Management Portal using the Analytics function.

To display system inventory information, complete the following steps:

1. In the Functions pane, click **Analytics**.
2. Click the **System Inventory** item.  
The **System Inventor** pane is displayed.
3. To sort the report by attribute, click the column heading for that attribute. For example, you can sort the report by memory size by clicking the **Memory Size** column heading. Clicking inside a heading changes the sort order for the attribute between ascending and descending order. You can also change column widths by clicking the heading row at any column border and dragging it to the left or right.
4. To export the contents of the table of results to a file, click **Save**.  
The **Export Report** window is displayed.
5. From the **File Type** list, select the file type to which you want to export the system inventory information.  
The available file formats are:
  - PDF
  - CSV
  - XML
6. To export the report, click **Export**.  
Your browser opens the exported file based on the file-handling preference specified for that format of file.

## Displaying software server inventory information

You can display a software server inventory report for all of your discovered configuration items (CIs) in the Data Management Portal using the Analytics function.

To display software server inventory information, complete the following steps:

1. In the Functions pane, click **Analytics**.
2. Click the **Software Server Inventory** item.  
The **Software Server Inventory** pane is displayed.
3. To sort the report by attribute, click the column heading for that attribute.  
For example, you can sort the report by version by clicking the **Version** column heading. Clicking inside a heading changes the sort order for the attribute between ascending and descending order. You can also change column widths by clicking the heading row at any column border and dragging it to the left or right.

**Note:** The data for the report is retrieved from the database one page at a time, and some displayed values are generated for display. Depending on the data included in a particular page, this can result in unexpected sorting results.

4. To export the contents of the table of results to a file, click **Save**.

The **Export Report** window is displayed.


5. From the **File Type** list, select the file type to which you want to export the software server inventory information.

The available file formats are:

- PDF
- CSV
- XML

**Note:** If you are using the Microsoft Internet Explorer browser and connecting using a secure session, you cannot export the report information to a file. The following alternatives are available:

- Use an alternative web browser.
- Use Tivoli Common Reporting for viewing and administering reports.

6. To export the report, click .

Your browser opens the exported file based on the file-handling preference specified for that format of file.

## Administration tasks

You can use the Data Management Portal to perform administration tasks.

### Creating users

When the file-based registry is used for user management, you can create a new user and assign roles to that user.

To create a user, complete the following steps:

1. Start the Data Management Portal.
2. Click **Administration > Users**.

A list of users is displayed.

3. Click **Create User**.

The **Create User** window is displayed.

4. Enter the following information for the new user in the following fields:

- Username
- Email address
- Session timeout (in minutes)

For an unlimited session timeout for the Discovery Management Console, the session timeout value is -1.

You can set a session timeout value for each user in the **Users** pane of the Data Management Portal, by setting the **Session Timeout** value (in minutes).

- Password (twice for confirmation)

**Note:** It is recommended to leverage TADDM integration with LDAP/Active Directory to enforce password policies for length, complexity and duration.

5. Assign roles to the new user.

For each role that you assign, perform the following steps:

- a) Select the check box for that role.

- b) Specify the scope of the role by selecting one or more access collections.
6. Click **Create User**.
- The user is added. The list of users is displayed again, with the new user included in the list.

## Editing users

When the file-based registry is used for user management, you can change the information for an existing user.

In addition to changing the user details (email address, password, and session timeout), you can also change the access permissions by assigning different roles and access collections.

To edit a user, complete the following steps:

1. Start the Data Management Portal.
2. Click **Administration > Users**.  
A list of users is displayed.
3. Click the user name that you want to edit, and then click **Edit**.  
The information for the user is displayed.
4. Change the user details as needed:
  - Email address.
  - New password (twice for confirmation).
  - New password expiration date. If a date is specified that is not valid, the expiration is set to 90 days from the current date.
  - Session timeout (in minutes).
5. Change the roles and access permissions to meet your security requirements.
6. The button that you click to save your changes depends on the properties that you change:
  - To save User Detail properties, click **Change**
  - To save Change Password properties, click **Change Password**
  - To save Change Role Assignment properties, click **Change Role**

## Deleting users

When the file-based registry is used for user management, you can delete a user that you created.

**Restriction:** The administrator cannot be deleted.

To delete a user, complete the following steps:

1. Start the Data Management Portal.
2. Click **Administration > Users**.  
A list of users is displayed.
3. Select the user you want to delete and click **Delete**.  
A confirmation window is displayed.
4. Click **OK**.  
The user is deleted.

## Creating user groups

When the file-based registry is used for user management, you can create a new user group.

To create a user group, complete the following steps:

1. Start the Data Management Portal.
2. Click **Administration > User Groups**.  
The **User Groups** pane is displayed.

3. Click **Create Group**.

The **Create Group** pane is displayed.

4. In the **Create Group** pane, select the users for your user group.
5. Assign roles to the new user group.

For each role that you assign, perform the following steps:

- a) Select the check box for that role.
- b) Specify the scope of the role by selecting one or more access collections.

6. Click **OK**.

The user group is added. The list of user groups is displayed again, with the new user group included in the list.

## Editing user groups

When the file-based registry is used for user management, you can change the information for an existing user group.

In addition to adding or removing users in a user group, you can also change the access permissions by assigning different roles and access collections.

To edit a user group, perform the following steps:

1. Start the Data Management Portal.
2. Click **Administration > User Groups**.

A list of user groups is displayed.

3. Select the user name of the user group that you want to change and click **Edit**.

The **Edit User Groups** pane is displayed.

4. Add or remove users from the user group.
  5. If the security requirements of the user group have changed, change the roles and access permissions as needed.
  6. Click **OK**.
- Your changes are saved.

## Deleting user groups

When the file-based registry is used for user management, you can delete a user group that you created.

To delete a user group, complete the following steps:

1. Start the Data Management Portal.
  2. Click **Administration > User Groups**.
- The **User Groups** pane is displayed.
3. Select the user group you want to delete and click **Delete**.
- A confirmation window is displayed.
4. Click **OK**.
- The user group is deleted.

## Creating roles

If the predefined roles are not sufficient for your needs, you can create a new one with the permissions that you choose.

To create a new role, complete the following steps:

1. Start the Data Management Portal.
  2. Click **Administration > Roles**.
- A list of roles is displayed.

3. Click **Create Role**.

The **Create Role** window is displayed.

4. Type a name for the new role, then select the permissions that you want to grant.

5. Click **Create Role**.

The list of roles is displayed again, with the new role included in the list.

## Editing roles

You can edit a role to set its permissions.

**Restriction:** The predefined roles (administrator, operator, and supervisor) cannot be edited.

To create a new role, complete the following steps:

1. Start the Data Management Portal.
2. Click **Administration > Roles**.
3. From the list of roles, select the role you want to edit and then click **Edit**.
4. In the Edit Role window, select the permissions you want to grant.
5. Click **OK**.

The list of roles is updated to show the changes.

## Deleting roles

You can delete a role that is no longer needed.

**Restriction:** The predefined roles (administrator, operator, and supervisor) cannot be deleted.

To delete a role, complete the following steps:

1. Start the Data Management Portal.
2. Click **Administration > Roles**.  
A list of roles is displayed.
3. Click **Delete** next to the role that you want to delete.

The role is deleted.

## Domain management tasks

Synchronization server deployment provides TADDM functionality for an entire enterprise, thus enabling several domains to be managed by a single synchronization server.

### Adding a domain to your enterprise

You can use the **Domain Summary** pane to add a new domain to your enterprise.

This task is available for synchronization server deployments only.

To add a domain to your enterprise, complete the following steps:

1. Click **Domain Management > Domain Summary**.  
The **Domain Summary** pane is displayed.
2. In the **Domain Summary** pane, click **New**.  
The **Add Domain** pane is displayed.
3. In the **Add Domain** pane, enter the domain information.
4. To apply the information you entered, click **OK**.



## Changing a domain in your enterprise

You can use the **Domain Summary** pane to update the information for an existing domain in your enterprise.

This task is available for synchronization server deployments only.

To change the domain information for an existing domain, complete the following steps:

1. Click **Domain Management > Domain Summary**.

The **Domain Summary** pane is displayed.

2. In the **Domain Summary** pane, select the Domain to be changed and click **Edit**.

The **Edit Domain** pane opens.

3. In the **Edit Domain** pane, update the domain information.

4. To ensure that the Data Management Portal can contact the domain using the information specified in the Domain Details section, click **Test Connection**.

A confirmation window is displayed to tell you whether the connection is successful.

5. If you changed the domain information and want to save it without exiting the **Edit Domain** pane, click **Save Changes**.

The changes are saved and the **Edit Domain** pane remains open.

6. If you changed the domain information and you want to save it and go back to the **Domain Summary** pane, click **Apply**.

The changes are saved and the **Domain Summary** pane is open.

## Deleting a domain from your enterprise

If you no longer need to collect information on a particular domain, you can use the **Domain Summary** pane in the Data Management Portal to delete that domain from your enterprise.

To delete a domain, you must log in to the Data Management Portal as a user that has the Admin runtime permission. When a domain is deleted from the TADDM database, any access collections that need to be deleted must be manually deleted using either the API or the **api.sh** script. If the domain that you are deleting has authorization policies for access collections that were synchronized to the synchronization server database, you must manually remove access to these access collections using the Data Management Portal.

This task is available for synchronization server deployments only.

To delete a domain from your enterprise, complete the following steps:

1. In the **Domain Summary** pane, select a domain.

2. To delete the domain, click **Delete**.

A prompt is displayed to confirm that you want to delete the selected domain.

3. Click **OK**.

The deletion of the domain might take a long time to complete, but it is performed as a background task so you can continue to use the Data Management Portal. Progress for the task is displayed in the **Domain** list, beside the domain you are deleting. When the deletion operation completes, the domain is deleted from your enterprise and removed from the Domain Summary table.

## Moving a domain to another synchronization server

Use the **Domain Summary** pane to move a domain from one synchronization server to another.

When moving a domain, first make sure it is up and running.

This task is available for synchronization server deployments only.

To move a domain from one synchronization server to another, complete the following steps:

1. On the Data Management Portal running on the first synchronization server, click **Domain Management > Domain Summary**.

The **Domain Summary** pane is displayed.

2. In the **Domain Summary** pane, click **Delete**. The domain is deleted from the first synchronization server.

**Important:** If you have a large database, this process can take a few hours. The **Domain Status** pane displays that the delete operation is in progress until it finishes.

3. On the Data Management Portal running on the second synchronization server, click **Domain Management > Domain Summary**. The **Domain Summary** pane is displayed.

4. In the **Domain Summary** pane, click **New**.

The **Add Domain** pane is displayed.

5. In the **Add Domain** pane, enter the domain information.

6. To apply the information you entered, click **OK**. The domain is moved to the new synchronization server.

After adding the domain, it must be fully synchronized to have its resources added to the synchronization server. The domain of the user interface must be restarted so that it can authenticate to the new synchronization server.

## Refreshing the domain information in the TADDM database

When the status of a domain in your enterprise has changed, you can display the updated information in the **Domain Summary** pane.

This task is available for synchronization server deployments only.

To update domain information in the TADDM database, complete the following steps:

1. Click **Domain Management > Domain Summary**.

The **Domain Summary** pane is displayed.

2. In the **Domain Summary** pane, select the domain for which you want information to be sent to the TADDM database.

3. Click **Refresh** in the **Domain Summary** pane.

The availability status of the domains in your enterprise that are updated in the TADDM database.

## Starting a Discovery Management Console

Although there is no Discovery Management Console for the synchronization server, you can access the Discovery Management Console of any of the domains in the enterprise from the synchronization server.

This task is available for synchronization server deployments only.

To start a Discovery Management Console for a domain in your enterprise, complete the following steps:

1. Optional: If you are using Firefox version 3.0 or later, make sure the TLS 1.0 encryption protocol is enabled in your browser settings.

To enable TLS 1.0 in Firefox 3.6:

- a) Click **Tools > Options**.
- b) In the Options window, click **Advanced**.
- c) Click the Encryption tab.
- d) Select **Use TLS 1.0**.

2. Click **Domain Management > Domain Summary**.

The **Domain Summary** pane is displayed.

3. Ensure that the domain for which you want to access the Discovery Management Console has an open padlock icon alongside the domain name, specifying that it accepts unsecured connections.

4. Select the domain for which you want to start the Discovery Management Console.

5. Click **Start**.

The Discovery Management Console for the selected domain is displayed.

## Starting a Discovery Management Console in secure mode

After secure connections have been configured, you can launch a secure connection to the Discovery Management Console for any of the domains in the enterprise from the Data Management Portal, running on the synchronization server.

Make sure your browser is configured to use a supported Java runtime environment, and that your computer meets all TADDM client hardware and software requirements. For more information, refer to the *TADDM Installation Guide*.

This task is available for synchronization server deployments only.

To configure SSL connection settings and start a Discovery Management Console in secure mode for a domain in your enterprise, complete the following steps:

1. Click **Domain Management > Domain Summary**.

The **Domain Summary** pane displayed.

2. In the **Domain Summary** pane, select the domain for which you want to start the Discovery Management Console in secure mode.
3. Optional: If necessary, configure the SSL connection settings.

You must complete this step in either of the following situations:

- The SSL connection settings have not been configured. This step must be completed before you can start the Discovery Management Console in secure mode. (If you have not yet configured the SSL connection settings, the **Start in Secure Mode** button is not enabled.)
- The list of managed domains has changed. In this situation, you must repeat the SSL connection settings configuration to update the truststore (especially if any domains have been added).

To configure the SSL settings, complete the following steps:

- a) Click **SSL Connection Settings**.

The **SSL Connection Settings** window is displayed.

- b) Click **Download Trust Store**. Take note of the directory to which you save the truststore file.

**Important:** Do not change the name of the truststore file.

The truststore file is downloaded using a secure connection.

- c) In the **Directory for the trust store** field, type the directory to which you saved the truststore file, without the trailing path separator.  
For example, if you saved the truststore file as

```
C:\domain_certs\Domain.cert
```

enter the directory for the truststore as

```
C:\domain_certs
```

- d) Click **OK**.

The truststore file information is saved in a browser cookie and is used when you start a secure connection.

To disable non-secure connections and force the use of SSL connections, set **com.collation.security.enforceSSL** to *true*. The default value for this property is *false*.

4. Optional: If you are using Firefox version 3.0 or later, make sure the TLS 1.0 encryption protocol is enabled in your browser settings.

To enable TLS 1.0 in Firefox 3.6:

- a) Click **Tools > Options**.
- b) In the Options window, click **Advanced**.
- c) Click the Encryption tab.
- d) Select **Use TLS 1.0**.

5. In the **Domain Summary** pane, ensure that you have selected the domain for which you want to start the Discovery Management Console in secure mode.
6. Click **Start in Secure Mode**. If you are prompted to accept a security certificate, you must do so. You are asked to accept a security certificate only the first time that you connect to the domain. Your browser attempts to open `confignia.jnlp`.

**Note:** If the **Start in Secure Mode** button is not enabled, you must configure the SSL connection settings as described in step “3” on page 103.

7. If your browser prompts you to specify how you want to use `confignia.jnlp`, do one of the following:
  - a) Open `confignia.jnlp` with Java Web Start.
  - b) Save the `confignia.jnlp` file locally. To launch the Discovery Management Console at a later time, open `confignia.jnlp` with Java Web Start.

The Discovery Management Console for the selected domain is initialized using SSL, and displayed.

## Specifying synchronization on demand

On demand synchronization can be done either through full-synchronization or incremental synchronization using the Data Management Portal running on a synchronization server.

With full synchronization, all configuration items in a domain are synchronized to the synchronization server database. With incremental synchronization, only configuration items that have changed on the domain are synchronized to the synchronization server database.

This task is available for synchronization server deployments only.

To specify synchronization between the synchronization server database and the domain server database, complete the following steps:

1. In the Functions pane, of the Data Management Portal, click **Domain Management > Synchronize**. The **Synchronize** pane is displayed.
2. In the Domains section of the **Synchronize** pane, click the domain that you want to synchronize. The **Synchronize** pane is displayed containing information on the selected domain.
3. In the On Demand Synchronization section, select **Perform full Synchronization** if you want to perform full synchronization between the synchronization server database and the domain server database. Otherwise, incremental synchronization is performed.
4. To start the synchronization immediately, click **Start**.
5. To stop the synchronization, click **Stop**.

## Specifying a scheduled synchronization

You can specify scheduled synchronization rather than performing on-demand synchronization using the Data Management Portal running on the synchronization server. Scheduled synchronization is always incremental, that is, only those configuration items that have changed on the domain are synchronized to the synchronization server database.

This task is available for synchronization server deployments only.

To set a specific time for synchronization to occur, complete the following steps:

1. In the Functions pane of the Data Management Portal, click **Domain Management > Synchronize**. The **Synchronize** pane is displayed.
2. In the Domains section of the **Synchronize** pane, click the domain that you want to synchronize. The **Synchronize** pane is displayed containing information on the selected domain.
3. In the Scheduled Synchronization section, click **New**. The **Schedule Period** pane is displayed.
4. Type the name of the schedule.
5. Type the time when you want the synchronization to start.

6. From the list in the **Repeat** field, select the interval at which to perform the synchronization. The options are hourly, daily, or weekly.  
The option that you select is displayed next to the input box in the **Every** field.
7. In the **Every** field, enter a numerical value or the Repeat specifications to pass between each synchronization.
8. To save the information, click **Add**.  
The **Synchronize** pane is displayed with the new information in the table that is in the Scheduled Synchronization section.
9. To return to the **Synchronize** pane without saving the information, click **Cancel**.

## Viewing synchronization status

You can view synchronization status and a log of synchronization for the domains in your enterprise using the Data Management Portal running on the synchronization server.

This task is available for synchronization server deployments only.

To view synchronization status for a specific domain, perform the following steps:

1. In the Functions pane of the Data Management Portal, click **Domain Management > Synchronize**.  
The **Synchronize** pane is displayed.
2. In the Domains section of the **Synchronize** pane, click the domain that you want to synchronize.  
The **Synchronize** pane is displayed containing information on the selected domain.
3. In Last Synchronization Time section, click **View Sync Details**.  
The **Synchronization Status** pane is displayed containing the synchronization status and a synchronization log.

## Deleting a scheduled synchronization

If you no longer want a scheduled synchronization to occur, you can delete that synchronization schedule using the Data Management Portal running on the synchronization server.

This task is available for synchronization server deployments only.

To delete a scheduled synchronization of a domain in your enterprise, complete the following steps:

1. In the Functions pane of the Data Management Portal, click **Domain Management > Synchronize**.  
The **Synchronize** pane is displayed.
2. In the Domains section of the **Synchronize** pane, click the domain that you want to synchronize.  
The **Synchronize** pane is displayed containing information on the selected domain.
3. From the table in the Scheduled Synchronization section, select the schedule that you want to delete.
4. Click **Delete**.  
The schedule for that domain is removed from the table.

## Displaying enterprise inventory information

Using the Data Management Portal running on the synchronization server, you can display an inventory report for either the local domain or any of the remote domains included in your enterprise.

This task is available for synchronization server deployments only.

To display an inventory summary, complete the following steps:

1. In the Functions pane, click **Analytics** and do one of the following:
  - To display an inventory summary for the local domain, click **Inventory Summary (local)**. The **Inventory Summary** pane is displayed.
  - To display an inventory summary for any of the remote domains in your enterprise, click **Inventory Summary (domain)**. The Domain section of the **Inventory Summary** pane is displayed, containing a

- Domain** list. If you selected **Inventory Summary (domain)**, click the domain (in the **Domain** list) for which you want to display the inventory summary. The **Inventory Summary** pane is displayed.
- In the **Inventory Summary** pane, click an entity in the **Component Type** or **Inventory Detail** column. The **Inventory Details** pane opens.
  - In the **Inventory Details** pane, select a component and click one of the following buttons depending on the type of information you want to display:
    - Click **Details** to display the **Details** pane for the selected component.
    - Click **Changes** to display the **Change History** pane for the selected component.
    - Click **Save** to save the Inventory Summary report for the selected component.
    - Click **Mark For Comparison** to add this component to the list of components to be compared. The **Component Comparison** pane is displayed containing the selected component. To continue with the component comparison, complete the following steps:
      - In the **Inventory Details** pane, click the second component to be compared.
 

**Important:** You do not need to close the **Component Comparison** window before you click another component in the **Inventory Details** pane.

The **Component Comparison** window is displayed containing the second selected component.
      - Repeat this step for all the components that you want to compare.
      - In the **Component Comparison** window, select the components that you want to compare and click **Compare**. The **Component Comparison** pane is displayed.
- Important:** The same data discovered on more than one remote domain is displayed only on inventory summary for one particular remote domain.

## Fix Pack 6 Data Access Portal

The TADDM Data Access Portal is the IBM® Tivoli Application Dependency Discovery Manager (TADDM) web-based user interface for viewing the data in a TADDM database. This user interface is available with domain server deployment, synchronization server deployment, and with each storage server in a streaming server deployment.

## Logging In

You must have valid login credentials to log in to the Data Access Portal. In case you are a new user, contact your admin team for credentials.

### Prerequisite:

- Valid log in credentials
- Compatible browser. The following browsers support TADDM Data Access Portal:

Table 16. Specialized topologies	
Browser	Version
IE	Latest (11.x)
Firefox	Latest (61.x)
Chrome	Latest (70.x)

- Open a web browser and type any of the following URLs:

#### For http

http://<PSS/SSS IP>:9430/dap

#### For https

https://<PSS/SSS IP>:9431/dap

**Note:** The Data Access Portal can be accessed on both primary or secondary storage services.

2. In the TADDM Data Access Portal page, enter your **user id** and **password**.
3. Click **Login**.

For log in related issue, see 'Troubleshooting' section.

## Signing out

To sign out, complete the following steps:

1. Go to navigation bar and click the **user profile**.
2. Click **Sign Out**.

## Dashboard pane

Dashboard displays all configuration items available in the TADDM CMDB through a pie chart. This chart provides flexibility to quickly view CI details. Each slice of the pie chart represents a configuration item category and can be identified by a corresponding call-out.

Following are the salient features of the pie chart:

- Each slice of the pie chart represents a configuration item.
- Each slice is illustrated with distinct colors.
- You can click each slice to explode and explore related details.
- Each slice has call-outs that describes the name of the configuration item and its count.
- Under the pie chart, configuration item links are listed.

### Chart customization

By default, all available configuration items are shown in the pie chart, and slice size depends on count of the configuration item. Hence, configuration items that have lower count are often not clearly visible in the pie chart. In this scenario, you can hide the configuration items that have large count to view the updated chart. Also, you can view the pie chart in full screen. In the full-screen mode, you can explore the enlarged pie chart where even the small slices are prominently visible.

To hide a configuration item, click the configuration item links under the pie chart. Similarly, you can unhide any configuration items.

### Exploring detail of a configuration item

To explore detailed information for a configuration item, click the corresponding slice in the pie chart. A new pie chart appears and displays detail of the configuration items. This new chart illustrates subcategories of the configuration item. You can click again on the slices of the new chart to explore further. The inventory list displays the related component and its associated details. You can click component links to view related detail.

**Note:** Components links are active only for supported configuration items. For other configuration items, components links are not active.

## Print and download

### Pie chart

In the dashboard window, you can download a pie chart in various formats such as JPEG, PNG, SVG, and PDF.

In the pie chart pane, click the menu icon to choose the format that you want to download. You can also print a chart directly.

### Inventory result

You can download inventory results in two output formats: CSV and PDF.

## Search

Search functionality queries on components that belong to various classes, such as computer system, application server and database server, and displays following details of components:

Display name, Label, Name, Full qualified domain name, Numeric address.

Using search operation, you can perform Suggestive search, Normal search, and Advance search.

### Performing suggestive search

The search field suggests list of components based on what you type in the search field. For example, if you type **ra**, configuration items(CI) starting with **ra** will be displayed in the drop-down list.

1. In the **Search** field, type initial one or two alphabets or digits (in case of IP address) of the component detail you want to search.

A suggestion list is displayed.

2. Optional: You can click one of the following categories to filter the search result.

**All:** Displays all matching result.

**App server:** Displays result related to App server.

**Computer System:** Displays result related to computer system.

**Databases:** Displays results related to databases.

3. From the list, Click the item you want to view.

4. Search result is displayed.

**Note:** By default, 10 results are displayed per page.

### Performing normal search

In normal search, you can quickly search a component by its name or IP address.

1. In the **Search** field, type the component name or IP address you want to search.

A List of matching components is displayed.

2. Click the item you want to view.

Alternatively, Click the Search icon.

3. Search result is displayed.

**Note:** By default, 10 results are displayed per page.

### Performing advance search

The advance search feature helps you to use specific criteria for searching configuration items. You can search configuration items using combination of keywords, their values, operators and wild character.

1. In the **Search** field, enter the criteria for search. Search criteria is a combination of keywords, its value and operators. For more information, see examples.

Following section provides list of available keywords and operators:

Keyword

Table 17.	
Keyword	Detail
Name	Display Name
IP	IP Address
FQDN	Fully Qualified Domain Name



Table 17. (continued)	
Keyword	Detail
Date	Last Modified Time

### Operator

Available Operators are (=) (<=) (>=) (&)

**Note:** Use \* as a wild character.

- Click the **Search** icon or press enter.
- Search result displays list of components matching your search criteria.

**Note:** By default, 10 results are displayed per page.

### Examples of search criteria and combination

1. name=taddm.

This will list results having "taddm" as the display name.

2. name=taddm.

This will list results which have the display name starting with "taddm" only.

3. name=\*taddm.

This will list results that have the display name ending with "addm" only.

4. name=\*taddm\*&fqdn=taddm.lab.com.

This will list results that have the display name containing taddm and have the value of fqdn as taddm.lab.com.

5. ip=aa.bb.cc.dd.

This will list results having "aa.bb.cc.dd" as the IP address

6. date>=2019-01-16

This will list results having Last Updated date greater than or equal to 2019-01-16

## Viewing component details

You can view various attribute values for a component.

To view the attribute details or values of a component, complete the following steps:

- In the **Search** field, type the component detail you want to search. For more result, see 'Search' section.
- From the **Search Results** list, click the component you want to view. Search result displays a table with following attributes:
  - Component Name
  - Component Type
  - Type
  - IP Address
  - Last Updated
  - Actions
- Under the **component** head, click the corresponding component link you want to view. A detail page displays all the attributes of the component.

## Comparing components

You can compare configuration items and view their similar and different attributes. Maximum five configuration items with same Type can be compared.

To compare the components, complete the following steps:

1. In the **Search** field, type the component details. A list of matching components is displayed.
2. From the **component** list, click a **component** you want to compare. A table of matching components are displayed.

**Note:** You can click any of the following categories to filter the search result:

**All**

Display all matching results

**App Server**

Display items related to App Server

**Computer system**

Displays items related to Computer System

**Databases**

Displays items related to Databases

**Network**

Displays items related to Network

**Cluster**

Displays items related to Cluster

3. You can compare components using either of the following ways:

- **Using Details Pane:**

- a. In the Search Result, click any component link that you want to compare. Detail pane is displayed.
- b. Click **Add to compare**.
- c. Click the **search result** link from the breadcrumbs. The Search Result pane is displayed.
- d. From the components list, click the **similar component** link and click **ADD** to compare.
- e. Go to **step 4**.

- **Using Search Result Pane:**

- a. In the **search result** pane, select the **check boxes** of the corresponding components that you want to compare.
- b. Click **Add to compare**.
- c. Go to **step 4**.

4. Click **View Comparison**. View comparison pane is displayed.
5. In the **View Comparison** pane, set the criteria for the comparison. For more details, see *View Comparison Pane section*.
6. **Optional:** If you want to remove any component from the comparison list, click the corresponding **Trash** icon.
7. Click **Compare Now** to start the comparison. Comparison Results pane displays the result.

## Viewing Relationship

This page shows relationship between classes of the same or different types. Each relationship has a definition or type. These different relationship types carry a certain semantic that pertains to the kind of association between the resource instances.

To view the relationship of a component, complete the following steps:

1. In the **Search** field, type the component detail (such as, component system, database server, application server or IP address) you want to search. A List of matching components is displayed.
2. From the **component** list, click a **component** you want to compare. Search result pane displays a table with following attributes:
  - **Component Name**
  - **Component Type**

- **Type**
- **IP Address**
- **Last Updated**
- **Actions**

3. Under the **Actions** head, click the **View Relationship** link of the component you want to view.

## Details pane

Details pane shows the attributes of the configuration item(CI). On the left side of the pane, available attributes are listed. You can click any attribute to view the corresponding details.

## View comparison pane

The Component Comparison pane contains the following options:

### Set As Key

Sets a component as a key to compare with the other component.

### Trash icon

Removes the component from the comparison list.

### UNDO

Adds the recently deleted component again to the comparison list.

### Level

Defines the depth of the comparison report, Deep option shows more details than Basic option.

### Basic

Select this option for general configuration information, such as port number settings, directories.

### Deep

Select this option for additional information about installed modules on Apache servers, deployed application objects (such as EJBs) and resources (such as JDBC and JMS) on WebLogic servers.

### Include Infrastructure Services

Infrastructure components are configured to communicate with numerous services, such as a DNS service or an NFS file system service.

If you select **Yes**, TADDM searches all the differences in service dependency between the compared CIs (Component).

**Note:** By default, this option is **NO**. If you have selected **Compute System** in the search page, this option will not be available.

### Include System

Include System compares the physical systems on which the software resides, in addition to comparing the server software.

If you select **Yes**, besides comparing the server software, search result will also show physical system on which software resides. For example, if you select **yes**, while comparing databases, TADDM compare database and the operating system.

If you select **No**, Search result will show only server software details.

**Note:** By default, this option is **NO**.

### Include Full List

You can view the comparison report in the following three ways:

**Similar:** you can see the similar attributes.

**Changes:** you can see the different attributes.

**Both:** you can see both similar and different attributes.

## Component Comparison: result pane

The Comparison Result pane displays tables with comparison details.

Similar and dissimilar attributes of the components are grouped separately. Each table comprises three columns, first column shows Attribute names, second column shows attribute values of the key component, and third column shows attribute values of the other component.

Color of the key component column is blue, and color of the other component column is green. By default, 10 results are displayed per page. At the bottom you can find total number of configuration items for your search. Based on **Include full list** selection, Comparison Result pane displays:

### Similar

Table with attributes that have similar values for key and non-key components.

Or

### Changes

Table with attributes that have different values for key and non-key components.

Or

### Both

Two tables, one with similar attribute values and another with different attribute values.

## Relationship pane

Relationship pane shows the following attributes:

### Relationship Name

- Represents the source instance relations with target instance and vice versa. For example, one of the relationship types in the CDM is **manages**, which represents the source instance participates in a controlling role to the target resource instance in the relationship

### Related CI

- Represents source or target instance in relation with selected CI

### Source or Target

- Each relationship instance has a source and a target, which are the relationship's roles. The number of instances that can take part in each role is important. Certain relationships only allow one instance to take part. Others allow any number of instances. The number of instances that can participate in each role is known as the cardinality of the relationship.

## User interface reference

---

The TADDM user interfaces are the Discovery Management Console and the Data Management Portal. The specific windows that are available in either of these user interfaces depend on the type of TADDM deployment.

### Discovery Management Console

The TADDM client user interface for managing discoveries. This console is also known as the Product Console. It is applicable to a domain server deployment and to discovery servers in a streaming server deployment. The function of the console is the same in both of these deployments.

### Data Management Portal

The TADDM web-based user interface for viewing and manipulating the data in a TADDM database. This user interface is applicable to a domain server deployment, to a synchronization server deployment, and to each storage server in a streaming server deployment. The user interface is very similar in all deployments, although in a synchronization server deployment, it has a few additional functions for adding and synchronizing domains.

## Discovery Management Console windows and controls

The Discovery Management Console displays information from a discovery server.

Table 18 on page 113 describes the items that can be viewed in the **Discovery** tab of the Discovery Management Console Functions pane.

Table 18. Items in the Discovery tab	
Discovery tab item	Description
Overview	Displays the current discovery status. You can run a discovery, monitor the discovery progress, and examine errors. You can also stop a discovery, view a discovery log, and view the scope log.
Scope	Describes the IP address range that is discovered during the discovery process. The IP address range is discovered by using IP addresses, IP address ranges, and subnets. You can modify the scope by adding, editing, or removing entries. Scope sets can be grouped in scope groups. You can modify scope groups by adding and removing scope sets.
Access List	Displays the access list defined for the server. An access list is a collection of user names, passwords, and Simple Network Management Protocol (SNMP) community strings used by the server to access discovery targets in your infrastructure. You can add, edit, and delete entries in the access list.
Custom Servers	Displays custom servers. You can add, edit, copy, or delete custom servers.
Computer Systems	Displays all templates created to customize the discovery of Computer Systems. The user can create templates to customize the discovery of a Computer System.
Anchors and Gateways	Displays the anchors and gateways in your infrastructure. Anchors are used to discover entities within restricted network areas without requiring a dedicated server in the area. Gateways are required for non-shallow discovery of Windows servers. They are not required for credential-less discovery (Discovery Profile 1) of Windows servers. You can add or delete anchors and gateways, as well as specify the communication port.
Schedule	Displays discovery schedules, which instruct server to run a discovery at a defined time. You can add new schedules, edit existing schedules, or remove a schedule as required.
History	Displays history information about past discoveries. You can display the activity and error messages that are associated with each discovery.
Discovery Profiles	Displays all the discovery profiles that are used during the discovery process.

## Overview pane

You can view discovery information in the **Overview** pane. You can view the **Overview** pane while a discovery is running.

The **Overview** pane contains the following information and buttons:

### Discovery ID

Displays the ID of each discovery currently running. To view information about a particular discovery, select the ID of that discovery from the list. When no discovery is running, the value **New Discovery** is displayed.

### Status

Displays the status of the discovery.

### Components Found

Displays the number of components discovered.

### Sensors Running

Displays the number of sensors that are running.

### Progress

A table with progress-related information, including a list of sensors and corresponding host names, IP addresses, and dates.

### Show only items of status

Filters the table by the status selected from the list.

### Information

Displays information about the selected sensor.

### Run Discovery

Prompts you to select a scope and profile with which to start a discovery.

### Scope Details

Displays information about the scope.

### Discovery Log

Prompts you to enter a string for which to search the discovery log.

## Scope pane

You can manage scope information using the Scope pane.

The **Scope** pane contains the information about scope sets and scope groups. The **Scope Sets** tab contains the following information:

### Scope sets

A logical collection of host names, IP addresses, ranges of IP addresses, or subnets.

### Method

Specifies whether to include or exclude the IP address, IP address range, or subnet.

### Type

The type of address specified, which includes the following values:

- Subnet-An IP subnet, such as 255.255.255.0
- Range-IP address range, such as 1.2.3.4 - 1.2.3.10
- Address-IP address, such as 1.2.3.4

### Value

The actual IP address, IP address range, or subnet.

### Description/hostname

A user-supplied description of the discovery scope.

### Add Set

Creates a scope set and adds it to the list of scope sets.

**Delete Set**

Deletes a scope set and removes it from the list of scope sets.

**Add**

Creates a scope and adds it to the table.

**Edit**

Edits the attributes of a scope.

**Delete**

Deletes a scope and removes it from the table.

The **Scope Groups** tab contains the following information:

**Scope groups**

The groups of existing scope sets. One scope set can be a member of several scope groups.

**Scope sets**

Specifies the list of scope sets that are members of a selected scope group.

**Add Group**

Creates a scope group and adds it to the list of scope groups.

**Delete Group**

Deletes a scope group and removes it from the list of scope groups.

**Add to group**

Adds scope sets to selected scope groups.

**Delete from group**

Deletes a scope set from scope group and removes it from the table.

**Add Scope window**

You can add a scope to a scope set and a new group of scope sets using the **Add Scope** window.

When creating a scope, you can specify an individual host, a subnet, or a range of IP addresses. To exclude individual hosts from the scope, add an exclusion for that host.

Go to **Discovery > Scope** and select the **Scope Sets** tab. The **Add Scope** window opens.

The **Add Scope** window contains the following information:

**IP Type**

Specify the format of the IP address or hostname you want to add. The valid values are:

- Subnet
- Range
- Host

**IP Address**

Specify the IP address in the appropriate format, depending on the **IP Type** value.

**Description**

Specify a description of the host.

**Add Exclusion**

Displays fields in which you specify the details of hosts you want to exclude from the scope.

**Remove**

Removes the exclusion.

When you want to add new group of scope sets go to **Discovery > Scope** and select the **Scope Groups** tab. The **Add Scope** window contains the following information:

**Scope Groups**

The list of scope groups

**Scope Sets**

The list of scope sets that are members of a chosen scope group.

To add scope sets to a scope group click **Add to Group** and select the scope sets that you want to add to a group.

## Access List pane

You can manage scope information using the **Access List** pane.

The **Access List** pane contains the following information:

### Type

The type of the access list entry.

### Name

The name of the access list entry.

### Username

The user name to log in to the component to be discovered.

### Scope Set Restrictions

Information about the scopes to which access to the component is restricted, if applied.

Table 19.	
Field	Description
<b>Add</b>	Add access details to the access list.
<b>Edit</b>	Edits existing access details.
<b>Delete</b>	Deletes existing access details.
<b>Move Up</b>	Moves the selected access details up one place in the access list.
<b>Move Down</b>	Moves the selected access details down one place in the access list.

## Access Details window

You can manage scope information using the **Access Details** window.

The **Access Details** window contains the following tabs:

- Access Information
- Scope Limitation

The fields displayed in the **Access Information** tab depend on the value selected from the Component Type list. The following table identifies the component types and the additional fields and lists that you are required to complete for the access list entry.

Table 20. Required component types, fields, and lists for access list entry	
Component Types	Fields and Lists
Application Server, Database, Messaging Servers	<b>Name</b> Name to identify the device in the access list. <b>User name</b> User name to access the server. <b>Password</b> Password to access the server. <b>Vendor</b> The vendor of the server or database.



Table 20. Required component types, fields, and lists for access list entry (continued)

Component Types	Fields and Lists
CSM Server	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>Password</b> Password to access the server.</p> <p><b>User name</b> User name to access the server.</p>
Cisco Device	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the device.</p> <p><b>Password</b> The password for the Cisco device, if you are using Telnet protocol, SSH1 or SSH2.</p> <p><b>Enable Password</b> The Enable password for the Cisco device, if you are using Telnet protocol, SSH1 or SSH2.</p> <p><b>Confirm Enable Password</b> The Enable password for the Cisco device, if you are using Telnet protocol, SSH1 or SSH2.</p> <p>The Cisco IOS sensor requires the SNMP sensor to be established and working against the device. If your Cisco IOS sensor is using a Telnet protocol and does not prompt for a user name, type default in the <b>User name</b> field.</p>
Cisco Works	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the server.</p> <p><b>Password</b> Password to access the server.</p>
Computer System, Computer System (Windows)	<p><b>Authentication Type</b> The type of authentication for the computer system.</p> <p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the computer system.</p> <p><b>Password</b> Password to access the computer system.</p>

Table 20. Required component types, fields, and lists for access list entry (continued)

Component Types	Fields and Lists
Computing Center Management System (CCMS)	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the SAP CCMS server.</p> <p><b>Client ID</b> The client ID of the SAP CCMS server.</p> <p><b>Password</b> Password to access the SAP CCMS server.</p>
High Availability Solutions	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the Veritas Cluster server.</p> <p><b>Password</b> Password to access the Veritas Cluster server.</p>
IBM Tivoli Monitoring	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the Tivoli Enterprise Portal Server.</p> <p><b>Password</b> Password to access the Tivoli Enterprise Portal Server.</p>
LDAP Service	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the LDAP server.</p> <p><b>Password</b> Password to access the LDAP server.</p>
Network Element (SNMP)	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>Community String</b> The community string for the network device.</p> <p><b>Confirm Community String</b> The community string for the network device.</p> <p>The SNMP Network element must be configured to answer queries from the TADDM server IP address.</p>

Table 20. Required component types, fields, and lists for access list entry (continued)	
Component Types	Fields and Lists
Network Element (SNMPV3)	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the device.</p> <p><b>Password</b> Password to access the device.</p> <p><b>Private Password</b> The password used if data encryption is set for SNMP.</p> <p><b>Authentication Protocol</b> The type of authentication protocol used by SNMP.</p>
SysImager Server	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the SysImager server.</p> <p><b>Password</b> Password to access the SysImager server.</p>
System Landscape Directory Server	<p><b>Name</b> Name to identify the device in the access list.</p> <p><b>User name</b> User name to access the System Landscape Directory Server.</p> <p><b>Password</b> Password to access the System Landscape Directory Server.</p>

The **Scope Limitations** tab contains the following information:

**Entire scope**

If selected, specifies that the access details are not limited by scope. This is the default value.

**Limit to selected scope sets**

If selected, specifies that the access details are limited to the selected scope sets.

**Limit to selected scope groups**

If selected, specifies that the access details are limited to the selected scope groups.

## Custom Servers pane

You can manage custom servers in the **Custom Servers** pane.

The **Custom Servers** pane contains the following information and buttons:

**Enabled**

Specifies whether TADDM acts on matches to this custom server template during discovery. Values are true or false.

**Icon**

The icon associated with the custom server.

**Name**

The name of the custom server.

**Type**

The type of custom server:

- AppServer
- Java EE Server
- Web Server
- Database Server

Each of the four generic custom server types listed above can be used to create a template for the discovery of server types that TADDM does not automatically categorize. During discovery, any unknown server is identified as a custom server of this type if its runtime information matches the criteria you specified in the template, and no other existing enabled sensor matches this criteria. These types may also run if the matching TADDM sensor runs and fails.

**Sensor-type servers**

The remaining Types correspond to the sensors that run in the environment. These sensor-type custom server templates run only in conjunction with an existing successful sensor, and are intended to extend discovery of a server by customizing certain aspects of how the sensor discovers the server. See Chapter 15 "Custom servers extensions" for more information.

If a credentialed sensor and a custom server both run for the same target during one or more discoveries, due to the different nature of the data discovered without credentials, objects created by the CST might not reconcile with sensor created artifacts.

**Action**

The action to perform during the discovery:

**Discover**

Adds the custom servers that match the template found during discovery in TADDM and removes the server from the unknown server report.

**Ignore**

Removes custom servers that match the template found during discovery from the unknown server report.

**Config Files**

The path to the configuration files with which the custom server is associated.

***Custom Server Details window***

You can create, edit, or view a custom server in the **Custom Server Details** window.

The **Custom Server Details** window contains the following tabs:

- General Info & Criteria
- Config Files

The **General Info & Criteria** tab contains the following information and buttons:

**Name**

The name of the custom server.

**Type**

The type of custom server:

- AppServer
- Java EE Server
- Web Server
- Database Server

Each of the four generic custom server types listed above can be used to create a template for the discovery of server types that TADDM does not automatically categorize. During discovery, any unknown server is identified as a custom server of this type if its runtime information matches the

criteria you specified in the template, and no other existing enabled sensor matches this criteria. These types may also run if the matching TADDM sensor runs and fails.

### Sensor-type servers

The remaining Types correspond to the sensors that run in the environment. These sensor-type custom server templates run only in conjunction with an existing successful sensor, and are intended to extend discovery of a server by customizing certain aspects of how the sensor discovers the server. See the "Developing custom server extensions" section in the *TADDM SDK Developer's Guide* for more information.

**Limitation:** You cannot use script-based sensors to create custom server extensions.

If a credentialed sensor and a custom server both run for the same target during one or more discoveries, due to the different nature of the data discovered without credentials, objects created by the CST might not reconcile with sensor created artifacts.

### Action

The action to perform during the discovery:

#### Discover

Adds the custom servers that match the template found during discovery in TADDM and removes the server from the unknown server report.

#### Ignore

Removes custom servers that match the template found during discovery from the unknown server report.

### Enabled

Specifies whether TADDM acts on matches to this custom server template during discovery. Values are true or false.

### Icon

The icon associated with the custom server.

**Note:** For some custom server types, the icon might not be displayed in the **Discovered Components** pane, the **Details** pane, and the **Custom Servers** pane of the Data Management Portal.

### Browse

Prompts you to select an image to use as the icon for the custom server.

### Identifying Criteria

Specify how you want the condition sets to be treated. The available choices are:

- All of the following conditions match (logical AND)
- Any of the following conditions match (logical OR)

### Criterion list

Specify the criterion type for the condition set. The valid types are:

#### Program name

The name of the executable program.

#### Window Service name

The name of a Window operating system service.

#### Argument

The arguments passed to the program.

#### Environment

The environment variables set for the program.

#### Port

The TCP port number on which the process is listening.

### Operator list

Specify the operator for the condition set. The valid operators are:

- is

- is-not
- contains
- does-not-contain
- starts-with
- does-not-start-with
- ends-with
- does-not-end-with
- regular-expression

The **Config Files** tab contains the following information and buttons:

**Filename**

Lists the configuration files with which the custom server is associated.

**Add**

Adds a configuration file to the table.

**Edit**

Edits the selected configuration file.

**Remove**

Removes the selected configuration file from the table.

**Text argument field**

Specify the text argument for the criterion.

**Remove**

Removes the criterion.

**Add Criterion**

Adds a criterion.

***Search Path for Capture File window***

You can add a configuration file to a custom server or a computer system in the **Search Path for Capture File** window.

The **Search Path for Capture File** window contains the following information and buttons:

**Type**

Specify the type of file to capture. The valid values are:

- Config File
- Software Module
- Application Descriptor Directory/File

**Search Path**

Specify the search path for the capture file. You must specify file details in two fields.

- In the first field, specify the path to the directory that you want to search. You can also select the following predefined values:

/

The root of the file system.

**\$PWD**

The current working directory of the running program.

**\$HOME**

The home directory of the user ID of the running program.

**C:**

A directory on your local computer.

**%ProgramFiles%**

The program files directory.

**%SystemRoot%**

The system root directory.

**Note:** The asterisk (\*) is not allowed in this field.

- In the second field, specify some information about the configuration file that you want to find. You can type the name of the file, the extension of the file, for example \*.txt, or an asterisk \* to search for all files in the specified directory.

**Capture file content**

Specify whether you want to capture the contents of the configuration file.

**Limit size of captured file to**

Specify the maximum number of bytes of the captured configuration file.

**Recurse Directory Search**

Specify whether you want to recurse through the directory structure, including all subdirectories, to search for the configuration file.

**Important:** If you want to capture file content, before you select the recurse option, make sure that the data that you want to capture is not too large. For example, check whether the subdirectories do not contain whole disks. Do not enter / in the first field and \* in the second field of the **Search Path**, because it might lead to out of memory errors and timeout. In case of large data, you can use the **Limit size of captured file to** option.

**Note:** In TADD 7.3.0.2, or earlier, the name of this check box is **Recurse Directory Content**.

**OK**

Saves the changes and closes the window.

**Cancel**

Discards the changes and closes the window.

## Computer Systems pane

You can manage computer systems using the **Computer Systems** pane.

The **Computer Systems** pane contains the following information:

**Enabled**

Specified if the computer system template is enabled during discovery.

**Icon**

Displays the icon representing the computer system template.

**Name**

Displays the name of the computer system template.

**Type**

Displays the type of the computer system template.

**Action**

Specifies whether the computer system should be discovered or ignored.

**Config Files**

Displays the configuration file associated with the computer system template.

**Save**

Saves the order of the computer system template list.

**Add**

Adds a computer system template.

**Edit**

Edits the selected computer system template.

**Copy**

Copies the details of the selected computer system template to a new computer system template. You are prompted for the name of the new computer system template.

**Delete**

Deletes the selected computer system template.

**Move Up**

Moves the selected computer system template up one place in the list of computer system templates.

**Move Down**

Moves the selected computer system template down one place in the list of computer system templates.

***Computer System Details window***

You can create, edit, or view a computer system in the **Computer System Details** window.

The **Computer System Details** window contains the following tabs:

- General Info & Criteria
- Config Files

The **General Info & Criteria** tab contains the following information and buttons:

**Name**

The name of the computer system.

**Action**

The action to perform during the discovery:

**Discover**

Adds the computer systems that match the template found during discovery in TADDM and removes the system from the unknown server report.

**Ignore**

Removes computer systems that match the template found during discovery from the unknown server report.

**Enabled**

Specifies whether TADDM acts on matches to this computer system template during discovery. Values are true or false.

**Icon**

The icon associated with the computer system.

**Browse**

Prompts you to select an image to use as the icon for the computer system.

**Operating System**

Specifies that the computer system is running an operating system.

**MIB**

Specifies that the computer system is a Management Information Base (MIB).

**Identifying Criteria**

Specify how you want the condition sets to be treated. The items displayed in this section depend on the type of the computer system.

If the computer system is running an operating system, you can specify only one criterion, based on the operating system running on the computer system.

If the computer system is a MIB, the available choices are:

- All Criteria
- Any Criteria

**Criterion list**

Specify the criterion type for the condition set.

The valid types for an operating system computer system are:

**Operating System**

The operating system of the computer system.



The valid types for a MIB computer system are:

**SysOID**

The SysOID value for the MIB computer system.

**Sys description**

The Sys description value for the MIB computer system.

**Operator list**

Specify the operator for the condition set.

The valid type for a operating system computer system is:

- is

The valid types for a MIB computer system are:

- is
- is-not
- contains
- does-not-contain
- starts-with
- does-not-start-with
- ends-with
- does-not-end-with
- regular-expression

**Text argument field**

Specify the text argument for the criterion. This button is enabled only when the computer system type is MIB.

**Operating system list**

Specify the operating system of the computer system. This list is enabled only when the computer system type is operating system. The valid values are:

- Linux
- SunOS
- AIX
- HP-UX
- Windows
- OpenVMS
- Tru64
- Darwin

**Remove**

Removes the criterion. This button is enabled only when the computer system type is MIB.

**Add Criterion**

Adds a criterion to the list of criteria. This button is enabled only when the computer system type is MIB.

The **Config Files** tab is enabled only when the computer system type is operating system. The **Config Files** tab contains the following information and buttons:

**Filenames**

Lists the configuration files with which the computer system is associated.

**Add**

Adds a configuration file to the table.

**Edit**

Edits the selected configuration file.

**Remove**

Removes the selected configuration file from the table.

**Anchors and Gateways pane**

You can specify information about anchors and Windows gateways in the **Anchors and Gateways** pane.

The **Anchors and Gateways** pane contains the following information and buttons:

**Type**

The type of anchor or gateway. The types include Windows gateway and anchor server.

**Address**

The IP address of the anchor or Windows gateway host.

**Port**

The port used for communication between the TADDM server and the anchor hosts.

**Scope Set**

Either use the entire discovery scope or limit the scope to the specified set.

**Add Anchor window**

You can create, edit, or view an anchor or Windows gateway in the **Add Anchor** window.

The **Add Anchor** window contains the following information:

**Type**

Displays the type of the anchor. The valid values are:

- Anchor
- Windows Gateway

**Set By**

Specify whether you want to set the anchor by IP address or hostname. Depending on the selected value, either the **Address** field, or the **Host Name** field is displayed.

**Address**

Specify the IP address of the anchor.

**Host Name**

Specify the host name of the anchor.

**Entire scope**

Specifies that the anchor can be searched for across all discovery scopes.

**Limit to selected scope sets**

Specifies that the anchor can be searched for only in a particular discovery scope. You are prompted to select the discovery scope sets.

**Limit to selected scope groups**

Specifies that the anchor can be searched for only in a particular discovery scope. You are prompted to select the discovery scope groups.

**Schedule pane**

You can view schedule information in the **Schedule** pane.

The **Schedule** pane contains the following information and buttons:

**Name**

Displays the name of the schedule.

**Next Discovery**

Displays when the next discovery will take place.

**Repeat Cycle**

Displays the length of the repeat cycle. The valid values are:

- None

- Hourly
- Daily
- Weekly
- Monthly

#### **Profile Used**

The type of profile that the discovery used. This can be one of the following options:

- Level 1 Discovery
- Level 2 Discovery
- Level 3 Discovery
- Custom

### ***Discovery Schedule window***

You can use the **Discovery Schedule** window to add a discovery schedule.

The **Discovery Schedule** window contains the following information and buttons:

The **Details** tab contains the following information and buttons:

#### **Name**

The name of the schedule.

#### **Start Time (server time)**

The start time of the discovery.

#### **Repeat**

The length of the repeat cycle. The valid values are:

- None
- Hourly
- Daily
- Weekly
- Monthly

When you select a repeat cycle, you can specify a numeric value for the cycle, for example, 3 hours, or 4 days.

The **Scope** tab contains the following information and buttons:

#### **Scope**

The discovery scope. In TADDM 7.3.0.3, and later, you can choose the following two options:

##### **Fix Pack 3 Dynamic content of selected scopes and groups**

You can select only scope sets and scope groups. In this mode, scope sets and scope groups are resolved to a list of elements just before a discovery. It means that you can modify the content of such scope sets and scope groups, and the defined schedule runs discoveries with up-to-date list of scope elements. As a result, you do not need to modify a schedule each time you change the content of a scope set, or a scope group.

##### **Fix Pack 3 Static, selected elements of scopes and groups**

You can select scope sets, scope groups, and single scope elements. The content of such scope is static, which means that only the chosen elements are discovered. If the scope set, or scope group content changes over time, the discovery is run against the elements that belonged to the scope at the time of the discovery schedule creation.

#### **Profile**

The type of the discovery profile that the discovery uses.

## ***Schedule Details window***

You can use the **Schedule Details** window to view the details of a discovery schedule.

### **Name**

The name of the schedule.

### **Start Time (server time)**

The start time of the discovery.

### **Frequency**

The frequency with which the discovery is run.

### **Profile Name**

The name of the discovery profile that the discovery uses.

### **Scopes**

The scope of the scheduled discovery.

**Note:** **Fix Pack 3** In TADDM 7.3.0.3, and later, this information is replaced with **Scope Sets**, and **Scope Elements**.

#### **Fix Pack 3 Scope Sets**

The list of scope sets and groups of the scheduled discovery. This information is available only when you select the **Dynamic content of selected scopes and groups** option in the **Scope** tab during the scope creation.

#### **Fix Pack 3 Scope Elements**

The list of scope elements of the scheduled discovery. This information is available only when you select the **Static, selected elements of scopes and groups** option in the **Scope** tab during the scope creation.

## **History pane**

You can view discovery history in the **History** pane.

The **History** pane contains the following information and buttons:

### **Start Time**

The date and time when the discovery started.

### **Completion Time**

The date and time when the discovery completed.

### **Completion Code**

The final status of the discovery.

### **Profile Used**

The type of profile that the discovery used. This can be one of the following options:

- Level 1 Discovery
- Level 2 Discovery
- Level 3 Discovery
- Custom

### **Scope Details**

Enables the display of information about the scope used. When the scope details are enabled, the following information is displayed:

#### **Sensor**

Displays the name of the sensor used.

#### **Host Name/IP**

Displays the host name or IP address of the target discovered.

#### **Date**

Displays the time and date of the discovery.

**Status**

Displays the status of the discovery.

**Description**

Displays information about the information discovered if the discovery completed, or error information if the discovery encountered problems.

**Information**

Displays overview information about the discovery, the target, and the information saved or errors received.

## Discovery Profiles pane

You can manage discovery profile information using the **Discovery Profiles** pane.

The **Discovery Profiles** pane contains the following tabs:

- Sensor Configuration
- Access Control
- Platform Properties

The **Discovery Profiles** pane contains the following information:

**Name**

The name of the discovery profile.

**New**

Adds a discovery profile.

**Save**

Saves the changes made to the selected discovery profile.

**Delete**

Deletes the selected discovery profile.

**Description**

Displays description information about the selected discovery profile.

The **Sensor Configuration** tab contains the following information about the selected discovery profile:

**Enabled**

Displays whether the sensor configuration is enabled or disabled.

**Sensor Name**

Displays the name of the sensor configuration.

**Type**

Displays the type of the sensor configuration.

**Scope Restrictions**

Displays scope restrictions associated with the sensor configuration, if any.

**Description**

Displays a description of the sensor configuration.

**Configure**

Configures the selected sensor configuration.

**New**

Adds a sensor configuration to the Sensor Configuration table.

**Delete**

Deletes the selected sensor configuration from the Sensor Configuration table.

**Clear All**

Clears all sensor configurations in the Sensor Configuration table.

**Select All**

Selects all sensor configurations in the Sensor Configuration table.

The **Access Control** tab contains the following information about the selected discovery profile:

**Type**

The type of the access list entry.

**Name**

The name of the access list entry.

**Username**

The user name to log in to the component to be discovered.

**Scope Restrictions**

Information about the scope sets and scope groups to which access to the component is restricted, if applied.

**New**

Adds access details to the access list.

**Edit**

Edits the selected access details.

**Delete**

Deletes the selected access details.

The **Platform Properties** tab contains the following information about the selected discovery profile:

**Included**

Specifies if the platform property set should be included.

**Name**

Displays the name of the platform properties.

**Value**

Displays the command specified by the platform properties.

**Scope Restrictions**

Displays scope restrictions associated with the platform properties, if any.

**Edit**

Edits the selected platform properties.

**Clear All**

Clears all platform properties.

**Select All**

Selects all platform properties.

**Create New Profile window**

You can create a discovery profile in the **Create New Profile** window.

The **Create New Profile** window contains the following information:

**Profile Name**

Type the name of the profile.

**Description**

Type a description of the profile.

**Clone existing profile**

If you want to create a new profile based on the properties of an existing profile you can clone an existing profile. From the list, select the profile that you want to clone.

**OK**

Saves the new profile and closes the window.

**Cancel**

Discards the changes and closes the window.

**Create Configuration window**

You can create a sensor configuration in the **Create Configuration** window.

The **Create Configuration** window contains the following information:

**Name**

Type the name of the sensor configuration.

**Description**

Type a description of the sensor configuration.

**Type**

Displays the type of the sensor configuration.

**Enable Configuration**

Specifies that the sensor configuration is enabled for use.

**Perform Script Based Discovery**

Specifies that script-based discovery is performed.

**Configuration table**

The following information about configuration properties is listed:

- Name
- Value

**No Restrictions**

Specifies that no scope restrictions apply to this sensor configuration.

**Add Restrictions**

Specifies that scope restrictions apply to this sensor configuration. You are prompted to select the scope that you want to apply to this sensor configuration.

**OK**

Saves the sensor configuration information and closes the window.

**Cancel**

Discards the sensor configuration information and closes the window.

***Edit Platform Properties window***

You can edit a platform property in the **Edit Platform Properties** window.

If you change the value of a property in this window, you must also select the **Included** check box to save that value. For example, if you change the value of the `com.ibm.cdb.discover.PreferScriptDiscovery` property to `true`, the value of `true` is not saved unless you also select the **Included** check box for the property.

The **Edit Platform Properties** window contains the following information:

***Property name***

Type the command used by the platform property.

**No Restrictions**

Specifies that no scope restrictions apply to this platform property.

**Add Restrictions**

Specifies that scope restrictions apply to this platform property. You are prompted to select the scope that you want to apply to this platform property.

## **Data Management Portal windows and controls**

The Data Management Portal displays information from a storage server.

### **Change Password menu item**

You can change your password in the Data Management Portal.

To change your password, click **File > Change Password**. Enter your current password, and then enter and confirm your new password.

Your new password becomes active the next time that you log on to the Data Management Portal.

## Details pane

The **Details** pane displays detailed information about a selected object.

You can export the information from the details pane, click the **Export Details** icon. You can export this information in XML, PDF file, or CSV format. Alternatively, click **File > Export Details** from the menu.

**Note:** The icon for some custom servers might not be displayed in the **Details** pane.

## Discovered Components pane

After you run a discovery, you can use the **Discovered Components** pane to view a list of discovered components and launch views with more detailed information about the components.

Using the **Discovered Components** pane, you can do the following tasks:

- Find a configuration item (CI) by navigating to it through the **Inventory Summary** link or the **Custom Query** link.
- Find a CI by typing all or some of the name in the **Filter** field. CIs, within the current category, that contain the filtered text are displayed. You cannot use wildcard symbols as part of the filter text.
- Page forwards or backwards through the list of displayed CIs. You can also specify the number of a particular page to display and configure the number of CIs to display in each page.
- Click the name of a CI to display information about it in the **Details** pane.
- Select the check box beside the CI name to select the CI. You can select one or more CI at a time.
- Select an action from the Actions list to perform an action on one or more selected CI.
- Click **Cart** to add selected CIs to the cart. By adding CIs to the cart, you can perform actions on multiple CIs whether they exist in the **Inventory Summary** hierarchy or are in the **Custom Query** list.
- Remove one or all CIs from the cart.
- Show dependent components, type of components in the dependent relationship, and see how the components were created.

**Note:** The icon for some custom servers might not be displayed in the **Discovered Components** pane.

The **Discovered Components** pane contains the following information:

### Cart

Displays the content of the cart. If the contents of the cart is currently being displayed, it returns to the discovered component view.

### Actions

If one or more CI is selected or the cart is not empty, the list contains valid actions. The actions displayed depends on the number and type of CI selected or contained in the cart. Some of the following actions are displayed:

#### Add to cart

Adds the selected CI or CIs to the cart.

#### Change History

Displays a pane for generating a Change History report. A Change History report identifies changes for components over a specified time.

#### Compare Across Versions

Displays a pane where you can generate a report that compares the selected component against previous versions of the same component.

#### Compare Components

Displays a pane where you can compare the selected components with other similar components.

#### Delete

Deletes the selected CI or CIs.

#### Edit

Displays the **Edit Component** window.



**Empty cart**

Empties the cart.

**Explore**

Displays, using the node centered topology wizard, a graph detailing the dependencies between CIs.

**Merge**

Merges the selected CIs.

**Rediscover**

Rediscovered the selected CI or CIs.

**Remove from cart**

Removes the selected CI or CIs from the cart.

**Show Dependencies**

Displays the **Dependencies** window.

**Show Details**

Displays detailed information about the selected component in the Details pane.

**Show Inventory Summary**

Displays the inventory summary for selected business applications or collections.

**Show System Connection Topology**

Displays a graphical report of any directly connected computer system.

**Show Topology**

Displays the topology graph.

**Show *ci\_type* Topology**

Displays a specialized topology for a particular CI type.

**Unknown Servers**

Displays a pane where you can view the unknown servers running on the computer system. You can define a custom server based on an unknown server so that it can be discovered.

**Custom Queries**

Contains saved custom queries. You can select one or more CI returned by a custom query.

**Inventory Summary**

Contains CIs, grouped in the following categories:

- App Servers
- Clusters
- Composites
- Computer Systems
- Custom Servers
- Database Servers
- Messaging Servers
- Network Elements
- Other Servers
- Services
- Storage
- Unknown IPs
- Web Servers

**Note:** With data-level security enabled, when you have restricted access to CIs, in the **Discovered Components** pane you can see only the CIs that you have access to, including business applications and collections.



Displays the first page.



Displays the previous page.

#### **Page number field**

You can enter the number of the page you want to display.



Displays the next page.



Displays the last page.



Displays the **"Change the page size"** window in which you can select the number of CIs to be displayed in each page.

### ***Unknown Servers pane***

After you run a discovery, you can use the **Unknown Servers** pane to view a list of unknown servers running on a computer system, and define a custom server based on an unknown server.

Unknown servers are identified after a discovery by a topology build agent. The topology build agent runs in the background on a periodic basis, depending on the value of the configured frequency, so unknown servers might not be recognized immediately after a discovery completes. Every four hours is the default frequency at which the topology build agent runs.

For this reason, if you run the Unknown Servers Report before the topology build agent has completed, the report might not list all unknown servers.

The **Unknown Servers** pane contains the following information:

#### **Details**

Displays the details of the selected unknown server.

#### **Create Custom Server**

Creates a custom server based on the selected unknown server.

#### **Delete**

Deletes the selected unknown server.

#### **Command**

The name of the executable program.

#### **Arguments**

The arguments passed to the program.

#### **Ports**

The TCP port number on which the process is listening.

#### **Service Name**

The name of a Window operating system service.

### ***Merge Component window***

After you run a discovery, you can use the **Merge Component** window to combine two or more configuration items (CIs) into one.

The **Merge Component** window contains the following information:

#### **Durable CI**

Displays the name of CI that is retained at the end of the merge operation.

#### **Transient CI(s)**

Displays the name of CIs that are deleted at the end of the merge operation.

**Move Up**

Raises the priority of the selected CI.

**Move Down**

Lowers the priority of the selected CI.

**Mark as durable**

Sets the selected CI as the durable CI.

**Display Name**

Lists the transient CIs selected for the merge operation.

***Dependencies window***

You can use the **Dependencies** window to create or delete a dependent component.

The **Dependencies** window contains the following information:

**New**

Displays the **Add dependency** window.

**Delete**

Deletes the selected dependent component.

**Dependent**

Displays the name of the other components in the dependency relationship.

**Type**

Displays the type of the other components in the dependency relationship.

**Created By**

Displays how the dependency was created either manually added or discovered.

***Add dependency window***

You can use the **Add dependency** window to create a dependency relationship.

Select the component that you want to create a relationship for in the Discovered Components pane and then select the target component from the **Add dependencies** window. There are two types of relationship, provider or dependent relationship.

The **Add dependencies** window contains the following information:

**Provider**

Click **Provider** and select a target component from the list. The initial component selected in the Discovered Components pane is now a provider to the selected target component.

**Dependent**

Click **Dependent** and select a target component from the list. The initial component selected in the Discovered Components pane is now dependent on the selected target component.

The following list shows the type of dependencies that you can have between components:

- AppServer -> AppServer
- AppServer -> AppServerCluster
- AppServer -> Service
- AppServer -> ComputerSystem
- ComputerSystem -> AppServer
- ComputerSystem -> AppServerCluster

**Discovery**

You can display domain summary information for all domains in your environment by clicking the items listed under **Discovery** in the **Functions** pane.

You can display the following types of discovery information:

- Scope information

- Custom server information
- Business entity information

### ***Scope pane***

You can manage scope information for the domains in your environment by configuring scope sets.

**Note:** To configure scope groups, refer to [“Scope pane” on page 114](#) in Discovery Management Console.

The **Scope** pane contains the following information:

#### **New scope set**

Creates a scope set and adds it to the list of scope sets.

#### **Delete scope set**

Deletes a scope set and removes it from the list of scope sets.

#### **Scope sets**

A logical collection of host names, IP addresses, ranges of IP addresses, or subnets.

#### **New**

Creates a scope and adds it to the table.

#### **Edit**

Edits the attributes of a scope.

#### **Delete**

Deletes a scope and removes it from the table.

#### **Method**

Specifies whether to include or exclude the IP address, IP address range, or subnet.

#### **Type**

The type of address specified, which includes the following values:

- Subnet-An IP subnet, such as 255.255.255.0
- Range-IP address range, such as 1.2.3.4 - 1.2.3.10
- Address-IP address, such as 1.2.3.4

#### **Value**

The actual IP address, IP address range, or subnet.

#### **Description/hostname**

A user-supplied description of the discovery scope.

### ***Custom Servers pane***

You can manage custom server information for the domains in your environment.

**Note:** The icon for some custom servers might not be displayed in the **Custom Servers** pane.

The **Custom Server** pane contains the following information:

#### **New**

Creates a custom server and adds it to the list of custom servers.

#### **Edit**

Edits the attributes of a custom server.

#### **Copy**

Creates a custom server based on an existing one.

#### **Delete**

Deletes a custom server from the list of custom servers.

#### **Move**

Changes the order in which custom servers are listed.

**#**

Displays the number of a custom server.

**Enabled**

Displays whether a custom server is enabled or not.

**Name**

Displays the name of a custom server.

**Type**

Displays the type of a custom server.

**Action**

Displays whether instances of a server are discovered or ignored.

**Config File**

Displays the associated configuration files, if any.

### ***Custom Server Details window***

You can use the **Custom Server Details** window to create or edit a custom server using the Data Management Portal.

The **Custom Server Details** pane contains the following tabs:

- General Info & Criteria
- Config Files

The **General Info & Criteria** tab of the **Custom Server Details** pane contains the following sections:

- General Server Information
- Identifying Criteria

The General Server Information section contains the following information and buttons:

**Enabled**

If selected, specifies that the custom server is enabled.

**Icon**

Displays an icon for the custom server.

**Browse**

Displays the available images that can be used as an icon for the custom server.

**Name**

Displays the name of the custom server.

**Type**

Displays the type of the custom server.

**Action**

Specifies whether the custom server should be discovered or ignored by discoveries.

The Identifying Criteria section contains the following information and buttons:

**Perform the action when**

Specify how you want the condition sets to be treated. The available choices are:

- All of the following conditions match (logical AND)
- Any of the following conditions match (logical OR)

**Criterion type list**

Specify the criterion type for the condition set. The valid types are:

**Program name**

The name of the executable program.

**Window Service name**

The name of a Window operating system service.

**Argument**

The arguments passed to the program.

**Environment**

The environment variables set for the program.

**Port**

The TCP port number on which the process is listening.

**Operator list**

Specify the operator for the condition set. The valid operators are:

- is
- is-not
- contains
- does-not-contain
- starts-with
- does-not-start-with
- ends-with
- does-not-end-with
- regular-expression

**Text argument field**

Specify the text argument for the condition set.

+

Adds a condition set.

-

Removes the condition set.

The **Config Files** tab of the **Custom Server Details** pane contains the following information and buttons:

**New**

Adds a configuration file.

**Edit**

Edits the selected configuration file.

**Copy**

Copies the selected configuration file.

**Delete**

Deletes the selected configuration file.

**Type**

Displays the configuration file type.

**Search Path**

Displays the search path of the configuration file.

**File Name**

Displays the file name of the configuration file.

**OK**

Saves changes and closes the window.

**Cancel**

Discards changes and closes the window.

***Search Path for Capture File window***

You can use the **Search Path for Capture File** window to specify a search path for a capture file by using the Data Management Portal.

The **Search Path for Capture File** window contains the following information and buttons:

**Type**

Specify the type of file to capture. The valid values are:

- Config File
- Software Module
- Application Descriptor Directory/File

**Search Path**

Specify the search path for the capture file. The valid values are:

**/**

The root of the file system.

**\$PWD**

The current working directory of the running program.

**\$HOME**

The home directory of the user ID of the running program.

**C:**

A directory on your local computer.

**%ProgramFiles%**

The program files directory.

**%SystemRoot%**

The system root directory.

**File Name**

Specify the path and file name of the capture file in the text box, or type \* (asterisk) to specify all files in the selected directory.

**Capture file content**

Specify if you want to capture the contents of the configuration file.

**Limit size of captured file to**

Specify the maximum number of bytes of the captured configuration file.

**Recurse Directory Search**

Specify if you want to recurse through the directory structure, including all subdirectories, to search for the file.

**Note:** In TADDM 7.3.0.2, or earlier, the name of the checkbox is **Recurse Directory Content**.

**OK**

Saves the changes and closes the window.

**Cancel**

Discards the changes and closes the window.

***Move Before/After window***

You can use the **Move Before/After** window to change the order in which custom servers are listed in the Custom Servers pane of the Data Management Portal.

The **Move Before/After** window contains the following information and buttons:

**Move**

Specify whether the custom server to be moved should be moved before or after the selected custom server.

**Custom server table**

Lists the custom servers in descending order.

**#**

Displays the position of the custom server.

**Name**

Displays the name of the custom server.

**OK**

Saves the changes and closes the window.

**Cancel**

Discards the changes and closes the window.

**Grouping Patterns pane**

You can create grouping patterns and view information about the patterns in your environment.

**Fix Pack 2** This pane was moved to the **Analytics** section of Data Management Portal in TADDM 7.3.0.2. To see the content of the Grouping Patterns pane, see [“Grouping Patterns pane” on page 151](#).

**Topology Agents Groups Status pane**

You can use the **Topology Agents Groups Status** pane to ensure that data coming from the most recent discovery is ready and has been processed by all of the appropriate topology agents.

The topology agents, which are gathered in groups, run at specified intervals. After a discovery has finished, you must wait for the agents to complete to ensure that all of the data has been processed. This processing includes cleaning up unreconciled database objects and incomplete topology relationships.

The information in the **Topology Agents Groups Status** pane is reset when TADDM is restarted.

Each group includes a different set of topology agents. The following groups are defined in TADDM by default:

**Fix Pack 2** **Discovery**

The discovery group run frequency is not defined. The group includes the following topology agents:

- CustomAppServerTopoAgent

The discovery group is defined by the following entry in the `collation.properties` file:

```
com.ibm.cdb.topobuilder.groupinterval.discovery=
```

**Note:** The default value in hours is empty, preventing the agent from running.

**Background**

The background group runs every four hours. The group includes the following topology agents:

- DatabaseServerCleanupAgent
- HostDependencyAgent
- L2Agent
- RuntimeCleanupAgent

The background group is defined by the following entry in the `collation.properties` file:

```
com.ibm.cdb.topobuilder.groupinterval.background=4.0
```

**Cleanup**

The group includes the following topology agents:

- ObjectsWithoutAliasesCleanupAgent
- PersobjCleanupAgent
- AliasesCleanupAgent

The properties for this group are described in the *Cleanup agents properties* topic in the TADDM *Administrator's Guide*.

The cleanup group is defined by the following entry in the `collation.properties` file:

```
com.ibm.cdb.topobuilder.groupinterval.cleanup=4.0
```

**Dependency**

The dependency group runs every 30 minutes. The groups includes the following topology agents:



- AppDefinitionAgent
- AppDescriptorAgent
- AppServerClusterAgent
- AppTemplateAgent
- CDPAgent
- CitrixAgent
- CMClusterDependencyAgent
- CMSDISAgent
- CompositeCreationAgent
- ComputerSystemConsolidationAgent
- ComputerSystemTypeAgent
- ConnectionDependencyAgent2
- DerivedAppToAppDependencyAgent
- DerivedSwitchToDeviceDependencyAgent
- DiscoveryLogCleanupAgent
- DNSDependencyAgent
- DNSServiceAgent
- DominoClusterDomainAgent
- DominoConnectionAgent
- ExchangeDependencyAgent
- ExchangeServer2007Agent
- GenericAppAgent
- HACMPDependencyAgent
- HISAgent
- HostStorageConnectionAgent
- IpNetworkAssignmentAgent
- J2EEServerDeploymentAgent
- JBossClusterAgent
- JDBCDependencyAgent
- LDAPServiceAgent
- MQServerAgent
- MSClusterAgent
- NFSDependencyAgent
- ObjectDisplayNameAgent
- OracleAppClusterAgent
- OracleDependencyAgent
- PortAppScanConsolidationAgent
- SAPDependencyAgent
- SoftwareHostReferenceAgent
- VCSDependencyAgent
- VmwareVirtualCSConsolidationAgent
- WebConnectionDependencyAgent
- WeblogicClusterAgent

- WebLogicServerDeploymentAgent
- WebSphereConnectionDependencyAgent

The dependency group is defined by the following entry in the `collation.properties` file:

```
com.ibm.cdb.topobuilder.groupinterval.dependency=0.5
```

### OnDemand

The OnDemand group results from the creation of a business application.

The OnDemand group is defined by the following entry in the `collation.properties` file:

```
com.ibm.cdb.topobuilder.groupinterval.bizapps=4.0
```

The **Topology Agents Groups Status** pane contains the following information:

### Integration

The integration group runs every six hours. The group includes OSLCAgent that integrates TADDM with Registry Services that is built upon the OSLC specification. It also includes OSLCAutomationAgent, which connects to OSLC Execute Automation Service Providers to download and create scope sets with IP addresses of other products' endpoints.

The time interval between runs is specified in the following entry in the `collation.properties` file:

```
com.ibm.cdb.topobuilder.groupinterval.integration=6.0
```

### Refresh

Refreshes the information in the **Topology Agents Groups Status** pane.

### Group Name

Displays the name of the topology agents group.

### Last Completion

Displays the time of the last completion.

### Finished

Displays whether the topology agent group has finished running. A green tick icon is displayed for a topology agent group that has finished. No icon is displayed if a topology agent group is running or has not been started since the most recent TADDM start.

## Topology

After running a discovery to capture information about your infrastructure, you can use the Data Management Portal to view graphical topology information for all domains in your environment by clicking the items that are listed under **Topology** in the **Functions** pane in the Data Management Portal.

The topology uses graphical objects to represent each component within a category and lines to represent the relationships between components. Icons are used to represent components.

The elements that are connected with the CI (configuration item) that you clicked last are highlighted in green.

The following table lists and describes the toolbar tool icons for topology views:













Table 21. Topology toolbar tool icons	
Tools icon	Description
	View an individual component within the topology graph.
	Select a group of components by clicking and dragging the selection rectangle.

Table 21. Topology toolbar tool icons (continued)	
Tools icon	Description
	Pan the topology graph. After you click the pan icon, use the cursor to drag the view to scroll around the topology graph when zoomed in.
	Zoom in on a rectangular section of the topology. After you click the zoom in rectangle icon, use the cursor to draw a rectangle on the graph to select the zoom region.
	Zoom in on the topology.
	Zoom out from the topology.
	Fit the entire topology within the workspace. This tool is useful for viewing all components without having to scroll.
	Show/hide the topology overview. The topology overview is a scaled copy of the entire topology graph. The portion of the topology graph currently being viewed is marked with a rectangle on the topology overview.
	Show changes. This button is used to highlight the components that changed. You can use the Highlight changes window to define the time frame that you want to use to check for changes.
	Export the topology to an image file.
	Highlight CoreCI. This button is used to highlight one or more core CIs.
	Customize the filter for CustomCollections (that is, Business Applications, Access Collections and Collections).

### **Business Applications window**

You can view business application topology graphs in the Physical Infrastructure window in the Data Management Portal.

To see a menu of available actions for an object in the topology, select and right-click the object.

The following table describes the available actions for business applications.

Table 22. Business applications pop-up menu items	
Menu item	Description
Add to cart	Adds the selected component to the cart.
Show Details	Displays detailed information about the selected component in the <b>Details</b> pane.
Show Topology	Displays the topology for the business application.
Edit	Displays the <b>Edit Component</b> window.
Delete	Deletes the selected component.
Rediscover	Rediscoveres the selected component.
Compare Across Versions	Displays a pane where you can generate a report that compares the selected component against previous versions of the same component.


Table 22. Business applications pop-up menu items (continued)	
Menu item	Description
Explore	Launches the node centered topology wizard, which you can use to create a node centered topology graph.
Change History	Displays a pane for generating a Change History report. A Change History report identifies changes for components over a specified period of time.
Compare Components	Displays a pane where you can compare the selected components with other similar components.
Filter	Displays the Filter pane, on which you can select to filter out types of elements.

#### Business Application Path details pane

You can view **Business Application Path** details pane in the business application topology.

To open **Business Application Path** details pane, on the graph, right-click a path's edge and choose the **Show Details** option.

The **Routes** tab contains detailed information about routes. Each path has at least one route. Routes can contain one or more segments. Details about each segment of a route are displayed in separate tables, which contain the following elements:

Table 23. Routes tab details.	
Item	Description
Source Object	The source object of the relationship, or dependency.
Relationship and Dependency	The type of a connection between nodes. Relationship is a basic type of a connection that results from Common Data model structure. Dependency represents a family of dependencies that can be customized and which exist as separate database elements. You can also define your own dependencies, see <a href="#">“Manually defining dependencies between configuration items”</a> on page 88.
Target Object	The target object of the relationship, or dependency.
 Traversal direction	The direction of the traversal of a specific relationship or dependency. For details, see <a href="#">“Traversing relations during pattern processing”</a> on page 212.

#### Related reference

[“Business application structure”](#) on page 177

Business application structure is created automatically, basing on a grouping pattern definition and on grouping pattern selectors definitions.

#### Export Topology window

You can export the currently displayed topology to an image file.

The **Export Topology** window displays the following information:

**File Type**

The type of file to which you want to export the topology. The following options are available:

- Portable Network Graphics (\*.png)
- Scalable Vector Graphics (\*.svg)
- JPEG format (\*.jpg)

**Height**

The height of the exported image.

**Width**

The width of the exported image.

**Export**

Exports the topology to an image file.

**Cancel**

Cancels the export of the topology.

***Filtering a topology***

You can add a filter to the displayed topology graph of a CustomCollection, that is, to Business Applications, Access Collections and Collections. You can filter out (hide) single or multiple nodes on a topology graph, or each node of a selected type.

**Restriction:** A filter remains selected (or deselected) during a session, which means it is not applied during start-up nor is it saved from one session to another.

**Tip:** When a topology graph displays connected nodes that have hidden (filtered) nodes between them, the connection between the nodes is a dashed line.

To filter a topology graph using the topology toolbar option, complete the following steps in the Data Management Portal:

1. In the Functions pane, click **Topology**.
2. Click **Business Applications**.

An overview topology for the item is displayed. If the displayed topology is not a CustomCollection, the filter button is disabled.

3. Click **Filter** on the topology toolbar.

The **Filter** window is displayed with all available filters selected.

4. Ensure that the type of filter or sub-filter that you want to apply is selected.

**Important:** Filter options are grouped by type. Selecting a filter option, for example 'Db2 Database', excludes all individual topology nodes of this type.

5. Click **OK** to view the filtered topology.

To filter a topology graph using the context menu:

6. While viewing the CustomCollection topology graph, select a node or a number of nodes to be filtered and right-click.

The Context menu is displayed.

7. Select **Hide nodes** or **Hide types**.

This hides selected node or nodes, or the whole group of nodes of the same type.

To display a whole, unfiltered topology graph:

8. While viewing the CustomCollection topology graph, disable the **Filter**.

**Analytics**

You can perform analytics and generate reports by clicking the items listed under **Analytics** in the Functions pane of the Data Management Portal.

You can perform the following tasks using the analytics or reporting functions:

- Run queries

- Display information about inventory, applications, change history, and services
- Generate custom and interactive reports

### ***Component Comparison pane***

You can run a Component Comparison report to gather information on the components in your domain.

The **Component Comparison** pane contains the following sections:

#### **Components**

This section contains information on the component on which you want to perform the component comparison.

#### **Options**

This section contains information on the type of comparison you want to perform.

The Components section contains the following information and buttons:

#### **Version**

The discovery version.

#### **Available**

The list of components in your environment. You can filter the components in the list based on the text typed in the **Filter** field. You can page forwards and backwards through the list of components by using the page forward and page back buttons.

#### **Included**

Components that you want included in the comparison report.

#### **Add**

Moves the selected components from the Available list to the Included list.

#### **Remove**

Moves the selected components from the Included list to the Available list.

#### **Set as Key**

Sets the component selected in the Included list as the key to compare all other components against.

The Options section contains the following information and button selections:

#### **Level**

The level of the component comparison report generated, from among the following levels:

##### **Basic**

Select this option for general configuration information, such as port number settings or directories.

##### **Deep**

Select this option for additional information about installed modules on Apache servers, and deployed application objects (such as EJBs) and resources (such as JDBC and JMS) on WebLogic servers.

#### **Include Infrastructure Services**

Infrastructure components are configured to communicate with various services, such as a DNS service or an NFS file system service. Selecting Include Services when comparing database components, for example, causes TADDM to find all differences in service dependencies among the components that are compared.

This option is off by default. It is unavailable in cases when the initial component selected is a computer system.

#### **Include System**

Selecting Include System compares the physical systems on which the software resides in addition to comparing the server software. Selecting Include System when comparing a database, for example, causes TADDM to compare the database and the computer system on which the database is running. When including systems in the component comparison, the systems must have the same type. If the systems have different types (for example, Windows and AIX), they cannot be included in the comparison.

This option is off by default. It is unavailable in cases when the initial component selected is a computer system.

### **Run Report**

Creates the Component Comparison report.

### ***Component Comparison: Results pane***

The **Component Comparison: Results** pane contains a comparison of the configuration attributes of two, or more, components.

The **Component Comparison: Results** pane contains the following sections:

#### **Configuration Entities**

A hierarchical list of all the common configuration attributes of the components that are included in the comparison.

#### **Components**

Two or more components, presented one per column, detailing the configuration and parameter information. You can scan the results to perform a quick comparison. The second column contains the values of the key component, with key values highlighted in the color gold. The rest of the columns represent the remaining components participating in the comparison.

Each row represents a configuration attribute for a given type of component. The key component value for an attribute is examined, and it is compared to the other component values, and creates a row if any differences are found using the following conventions:

- Values that are different from the key are highlighted in the color red.
- When a value is the same as the key value, the cell is left blank.
- When all components match the key value, the row is not displayed at all. A row is created only when it finds a difference.
- When a cell value in any component other than the key contains [Not Set], this attribute was set in the key but not in the particular component. Conversely, when the key contains [Not Set] for an attribute, it means that it was set for other components but not in the key component.
- To display a dialog box highlighting specific changes, you can click the values that are underlined.

A checksum value is calculated when it finds differences in configuration file content compared to the key, and puts that number (a link, highlighted in the color blue) into the results table.

The application does not list the actual file contents in the results. Clicking a checksum number causes an on-demand file comparison, creating a line-by-line comparison of the configuration file contents.

### ***Change History pane***

You can display a change history for a single domain environment or for all domains in your synchronization environment in the Data Management Portal.

The **Change History** pane contains the following tabs:

- Change History
- Results

The **Change History** tab contains the following sections:

- Timeframe
- Components

**Note:** Changes made to the members of custom collections are not displayed in the Change History report. For more information, see the *Data Management Portal problems* topic in the TADDM *Troubleshooting Guide*.

The Timeframe section in the **Change History** pane contains the following information:

**Set Date By**

Specify the type of timeframe you want. The options are:

**Relative Timeframe**

You can specify the Change History report time period by entering the number of months, weeks, days, or hours. The Change History report contains data that begins with the number of months, weeks, days, or hours that you specify and ends with data from the present time.

**Absolute Timeframe**

You can specify the precise date and time delimiting the start and end of the change history report.

**From**

The date and time from which you want to start the change history report.

**Timeframe input box**

Specifies the amount of time for which to show changes. Options are months, weeks, days, or hours.

**Start Date**

You can specify the precise date delimiting the start of the change history report.

**Start Time**

You can specify the precise time delimiting the start of the change history report.

**End Date**

You can specify the precise date delimiting the end of the change history report.

**End Time**

You can specify the precise time delimiting the end of the change history report.

The Components section in the **Change History** tab contains the following information:

**Component Type**

You can select components of the infrastructure software, infrastructure services, network tier, and system tier.

**Available Components**

Lists all the available components of the chosen component type you can choose from.

**Add**

Adds the selected component to the Included Components list.

**Remove**

Removes the selected component from the Included Components list.

**Included Components**

Lists the included components that you have already chosen.

**Run Report**

Generates a change history report based on the chosen options.

The table of changes in the **Results** tab contains the following information:

**Type**

The type of component that changed.

**Component**

The identifier for the specific component that changed.

**Change**

The change action, which can be one of the following actions:

- Created
- Updated
- Deleted

**Date**

The date and time when the change was detected by the TADDM discovery.



**Attribute**

The component attribute that changed.

**Old**

The value before the change occurred.

**New**

The value after the change occurred.

**Export**

Exports the results to a file. You can save the results as a .pdf, .csv, or .xml file.

***Change History pop-up window***

You can display a change history for a selected component in the Data Management Portal.

The **Change History** pane contains the following sections:

- Timeframe
- A table of changes

The Timeframe section in the **Change History** pane contains the following information:

**Set Date By**

Specify the type of timeframe you want. The options are:

**Relative Timeframe**

You can specify the Change History report time period by entering the number of months, weeks, days, or hours. The Change History report contains data that begins with the number of months, weeks, days, or hours that you specify and ends with data from the present time.

**Absolute Timeframe**

You can specify the precise date and time delimiting the start and end of the change history report.

**From**

The date and time from which you want to start the change history report.

**Timeframe input box**

Specifies the amount of time for which to show changes. Options are months, weeks, days, or hours.

**Show Changes**

Displays the changes as specified.

The table of changes in the **Change History** pane contains the following information:

**Type**

The type of component that changed.

**Component**

The identifier for the specific component that changed.

**Change**

The change action, which can be one of the following actions:

- Created
- Updated
- Deleted

**Date**

The date and time when the change was detected by the TADDM discovery.

**Attribute**

The component attribute that changed.

**Old**

The value before the change occurred.

**New**

The value after the change occurred.

**Custom Query pane**

You can manage custom query information in the **Custom Query** pane.

The Custom Query pane contains the **Saved Query** tab and depending on the tasks you are performing, it may contain some of the following additional tabs:

- New Query
- Edit Query
- Results

The **Saved Query** tab of the **Custom Query** pane contains the following information:

**New**

Creates a custom query and adds it to the list of custom queries.

**Edit**

Edits the attributes of the selected custom query.

**Copy**

Creates a custom query based on the selected one.

**Delete**

Deletes the selected custom query from the list of custom queries.

**Run Query**

Runs the selected custom query.

**Name**

Displays the name of a custom query.

**Description**

Displays a description of a custom query.

The **Results** tab of the **Custom Query** pane contains the following information:

**Details**

Displays the details for the component.

**Explore**

Displays the node centered topology for the component.

**Changes**

Displays the change history for the component.

**Mark For Comparison**

Adds the selected component to the list of components to be compared.

**Add to cart**

Adds the selected components to the cart in the **Discovered Components** pane.

**Save**

Exports a report to a PDF, CSV or XML file.

**Inventory Summary pane: Inventory tab**

You can display component details on the **Inventory Details** pane in the Data Management Portal.

The Inventory tab on the **Inventory Summary** pane contains the following information:

**Refresh icon**

Updates the pane with the latest data from the server.

**Filter icon**

Specify what conditions to apply to the inventory view, the choices are:

- All components (Shows all the components.)
- Active components (Shows components that have been updated since a specified date and time.)

- Dormant components (Shows components that have not been updated since a specified date and time.)
- Placeholders (Shows shallow configuration items that use ManagedSystemName as a naming rule.)

For active and dormant components, you must select a date and time. The component is active if it has been updated after the selected time. The component is dormant if it has not been updated after the selected time. Click **Apply** to apply the filter settings to the view.

### **Component Type**

The type of component, including web servers, application servers, database servers, and systems.

### **Inventory Detail**

A summary of the number of each component type. Click the component type icon to open the details tab for the component.

### ***Inventory Summary pane: Details tab***

You can display detailed inventory information for the systems in your enterprise.

You can navigate to this pane from the **Inventory Summary** pane.

The details tab on the **Inventory Details** pane contains the following information:

#### **Display Name**

The name of the application component or inventory component.

#### **Parent**

The type of component.

#### **Last Updated**

The date and time when the component was last modified.

#### **Details**

Displays details on the corresponding application or inventory component.

#### **Explore**

Displays the node centered topology for the corresponding application or inventory component.

#### **Changes**

Displays the change history for the corresponding application or inventory component.

#### **Save**

Creates a file containing the information displayed in the **Inventory Details** pane.

#### **Mark For Comparison**

Adds the selected component to the list of components to be compared.

#### **Add to cart**

Adds the selected components to the cart in the **Discovered Components** pane.

#### **Delete**

Deletes a selected row from the report. Displays the Delete Component window that shows all selected items, and their dependencies. From here, you can select and confirm the items you want to delete.

#### **Delete All**

Deletes all rows from the report.

After you confirm the deletion, the process deletes all the components. This task can take a long time to process and is carried out as a background, asynchronous task. To confirm that the components are deleted, click the **Refresh** icon to view the current inventory.

### ***Grouping Patterns pane***

You can create grouping patterns and view information about the patterns in your environment.

The information that you specify in the Grouping Patterns pane is used by BizAppsAgent as an entry criteria to automatically create grouping patterns.

**Note:** **Fix Pack 2** The Grouping Patterns pane is in the **Analytics** section of Data Management Portal in TADDM 7.3.0.2, and later. If you use an earlier version of TADDM 7.3 release, this pane is in the **Discovery** section.

**Note:** **Fix Pack 3** To display the Grouping Patterns pane with its content, you must have the Update permission granted for the DefaultAccessCollection.

## Grouping Patterns options

The Grouping Patterns pane contains the following options:

### New...

Creates a grouping pattern and adds it to the list of grouping patterns.

### Edit...

Edits the attributes of a grouping pattern.

### Copy

Copies the selected grouping pattern. By default, the new pattern is named *Copy of name of the copied pattern* but you can change it. By default, the copy is not enabled for processing.

### Delete

Deletes a grouping pattern and removes it from the list of grouping patterns.

### Execute

Starts the grouping pattern at the nearest possible time without having to wait for the scheduled execution time. This option is enabled only for the patterns that are enabled.

### Stop

Stops the patterns that are currently running. This option is enabled only for the patterns which have the Execution status *In progress*.

### Refresh View

Refreshes the list of grouping patterns, including the execution status information.

### Enable

Enables or disables the selected or all grouping patterns.

### **Fix Pack 2** Filter...

Allows you to filter grouping patterns by their names. Type the whole name, or a part of the name of the pattern that you want to find. The search is case sensitive. The following wildcard characters are allowed:

- \* - matches any number of occurrences of any character, it is automatically added at the end of your query
- ? - matches one occurrence of any character

For example, if you want to display all patterns that have *template* in their names, type *\*template*. The results contain the following types of names:

- The name starts with *template*, for example *template\_pattern*.
- *template* is in the middle of the name, for example *custom server template windows*.
- The name ends with *template*, for example *comp\_sys\_template*.

## Grouping Patterns table information

The Grouping Patterns table contains the following information:

### Name

Displays the name of a grouping pattern.

### Type

Displays the type of a grouping pattern. The valid types are Business Application, Access Collection, and Collection.

**Enabled**

Enables the grouping pattern to be processed. If this check box is not selected, the pattern is not enabled for processing.

**Description**

Displays a description of a grouping pattern.

**Last execution**

Specifies the last time when the pattern was executed.

**Next execution**

Specifies the next time when the pattern is executed.

**Execution status**

Specifies the status of the execution. For example, if a pattern is being processed, the status is **In progress**. If the pattern is set to be excluded from processing, the status is **Not enabled**.

*Create a new Grouping Pattern wizard*

Using the **Create new Grouping Pattern** wizard, you can create a grouping pattern of type Business Application, Access Collection, or Collection.

Depending on the specific details of the grouping pattern that you are creating, some or all of the following pages are displayed in the **Create a new Grouping Pattern** wizard:

- General Information
- Selectors
- Administrative Information

The General Information page contains some, or all, of the following information:

**Name**

Displays the name of the grouping pattern.

**Pattern type**

Displays the type of the grouping pattern.

**Fix Pack 1 Compatibility type**

This option is enabled only if the `com.ibm.cdb.serviceinfrastructure.earlier.ver.compatibility` property is set to `true`. It displays the type into which the old grouping entity is converted. The compatibility type is based on a chosen **Pattern type**:

- Collection - only Collection compatibility type is available.
- Access Collection - only Access Collection compatibility type is available.
- Business Application - the following types are available:
  - Business Application
  - Business Service.

**Schedule**

Displays the schedule of the grouping pattern. If you created your own schedules, they are on the list.

**Configuration**

Displays the configuration of the grouping pattern. If you created your own configurations, they are on the list.

**Description**

Displays a description of the grouping pattern.

**URL**

Displays a URL associated with the grouping pattern.

The Selectors page contains some, or all, of the following information:

**New**

Creates a rule. When you click **New**, the following information is available:

- **Selector name**  
Specifies a name for the new selector.
- **Fix Pack 1 Tier name**  
Specifies the name of the tier. When you enter any name in this field, all the objects that are found by the selector are added to this tier. The tier name that you enter here has precedence over the tier names that are specified in the tier configuration.
- **Description**  
Description of the selector.
- **Selection rule previews**  
Displays the query that is used to select core CI and grouping name expression. When you click **Choose...**, the following information is available:
  - **Selection type**  
Defines the selection type. You can choose between the three types of selection, MQL query selection, SQL query selection and Instance-based selection.
  - **Query**  
Defines the query which selects core CIs, or the elements that you selected manually as core CIs.
  - **Grouping Name Expression**  
Defines the expression which generates collection identifier.
  - **Test sample size**  
Defines the number of core CIs that are included in the test. This option is useful when your query finds many core CIs. By limiting the number of core CIs, the test does not take a long time to finish.
  - **Test**  
Tests whether the chosen selection type and grouping name expression are valid.

#### **Delete**

Deletes the selected selector.

#### **Disabled**

Creates a draft version of a selector. Disabled selectors are not used by BizAppsAgent in building custom collections.

#### **Data Traversal Template**

Defines how the selector traverses dependencies.

The **Use Depenedency Traversal Template** option enables data traversal.

The **Higher Up** and **Lower Down** options determine whether CIs connected upwards or downwards the dependency chain of the core CI are added to the generated custom collection.

The **Higher Down** and **Lower Up** options determine whether CIs connected downwards or upwards the dependency chain of already added CIs are also added to the generated custom collection.

The Administrative Information page contains some, or all, of the following information:

#### **Admin contact**

Displays an administration contact for the grouping pattern.

#### **Escalation contact**

Displays an escalation contact for the grouping pattern.

#### **Tracking number**

Displays a tracking number for the grouping pattern.

#### **Site**

Displays site information for the grouping pattern.

#### **Group name**

Displays a group name for the grouping pattern.

#### **Notes**

Displays notes for the grouping pattern.

## Related tasks

[“Creating grouping patterns” on page 179](#)

You can create new grouping patterns in the **Grouping Patterns** pane, in the Data Management Portal.

## ***Application Summary pane***

You can display applications summary information for the domains in your environment in the Data Management Portal.

The **Application Summary** pane contains the following information:

### **Save**

Exports a report to a PDF, CSV, or XML file.

### **Application Name**

The name of the business application.

### **Domain Name**

The domain name where the business application is located.

### **Groups**

Displays information associated with the functional groups in your environment.

### **Changes**

Displays the change history for your business application.

### **Details**

Displays details on your business application.

### **Explore**

Displays the node centered topology for the application.

### **Software Topology**

Displays the software topology for the application.

### **Physical Topology**

Displays the physical topology for the application.

### **Inventory**

Displays the inventory summary for the application.

### **Mark For Comparison**

Adds the selected component to the list of components to be compared.

### **Add to cart**

Adds the selected components to the cart in the **Discovered Components** pane.

## ***System Inventory pane***

The System Inventory report show details for all computer systems in your environment. It contains information from the Details pane. You can export this report in XML, PDF file, or CSV format. When you export this report, you export only the information that is displayed in the current view.

The **System Inventory** pane contains the following information:

### **Name**

The name of the component.

### **Model**

The model of the component.

### **CPU Type**

The type of processor installed on the component.

### **CPU Count**

The size of the processor.

### **Memory Size**

The amount of memory in the component in bytes.

**Manufacturer**

The manufacturer of the component.

**Save**

Exports a report to a PDF, CSV, or XML file.

**Add to cart**

Adds the selected components to the cart in the **Discovered Components** pane.

This report supports pagination.

***Software Server Inventory pane***

The Software Server Inventory report show details for all software for the application running on the component.

The **Software Server Inventory** pane contains the following information:

**Save**

Exports a report to a PDF, CSV, or XML file.

**Name**

The name of the component.

**Type**

The type of the component such as Db2Instance, OracleInstance.

**Version**

The version of software running on the component.

**Display Name**

The name of the component as it is displayed in the Discovered Components list.

**Add to cart**

Adds the selected components to the cart in the **Discovered Components** pane.

***BIRT Reports pane***

You can add, download, delete, and run BIRT reports in the BIRT Reports pane of the Data Management Portal.

The **BIRT Reports** pane contains the following information:

**Name**

Displays a report name.

**Description**

Displays a report description

**Run Report**

Displays a report that has been deployed to the TADDM BIRT runtime engine.

**New**

Deploys a new report design file to the TADDM BIRT runtime engine.

**Delete**

Deletes a report design file that has been deployed to the TADDM BIRT runtime engine.

**Refresh**

Reloads the list of deployed BIRT reports.

**Download**

Downloads a deployed report design file from the TADDM server to your computer for editing or cloning.

**Administration**

You can manage the administration information for the domains in your environment in the Data Management Portal console.

The following administration information can be managed for the domains in your environment:



- User Groups
- Users
- Roles

### ***User Groups pane***

You can display administration user information for your enterprise.

The **User Groups** pane displays the following information:

#### **Group Name**

The name of the group of users.

#### **Users**

The name of the user in the group.

#### **Roles**

The roles granted to the user in the group.

#### **Create Group**

Creates a new user group and adds that user group to the Domain Database table.

#### **Delete**

Deletes a user group and remove that user group from the Domain Database table. This button is available to users logged in as administrator or users with a role that has administrator permission.

#### **Edit**

Changes a user group's password. This button is available to users logged in as administrator or users with a role that has administrator permission.

### ***Create User Group and Edit User Group windows***

You can create or edit a user group for the users within each domain in your environment.

The **Create UserGroup** window and the **Edit UserGroup** window contain the following sections:

- General Information
- User Group Assignment
- Role Assignment

The General Information section displays the following information:

#### **Group Name**

The name of the user group. In the **Edit UserGroup** window, this field contains the name of the user group that you are editing.

The User Group Assignment section displays the following information:

#### **Available Users**

A list of users that can be in this user group.

#### **Add**

Adds a user to the user group.

#### **Remove**

Removes a user from the user group.

#### **Included Users**

List of users selected to be in the user group.

The Role Assignment section displays the following information:

#### **Assign**

The check box for assigning a role.

#### **Role Name**

The name of the role for this user.

#### **Permissions**

The name of type of permissions for this user.

## Access Collections

The check box for specifying access collections.

## Users pane

You can display administration user information for your enterprise in the **Users** pane of the Data Management Portal.

The information displayed depends on the type of authentication you are using, as set in the **com.collation.security.usermanagementmodule** property.

If you are using the default file-based authentication, all users are displayed.

If you are using LDAP or VMM-based authentication, the **Users** pane has a search field and users are displayed based on the search criteria entered. To list all users (up to the search limit set by the `collation.properties` file), in the search field, enter the "\*" character.

The **Users** pane displays the following information:

### User

The name of the user.

### Roles

The roles granted to the user.

### Email Address

The e-mail address of the user.

### Session Timeout

The session timeout value (in minutes) assigned to this user.

### Create User

Creates a new user and adds that user to the table.

### Delete

Deletes a user and remove that user from the Users table. This button is available to users logged in as administrator or users with a role that has administrator permission.

### Edit

Changes a user's password. This button is available to users logged in as administrator or users with a role that has administrator permission.

## Create User window

You can create users for the domains within your environment in the **Create User** window of the Data Management Portal.

The **Create User** window contains the following sections:

- General Information
- Role Assignment

The General Information section displays the following information:

### Username

The name of the user.

### Email Address

The e-mail address of the user.

### Password

The password for the user account.

### Confirm Password

The confirming password for the user account.

### Session Timeout (Mins)

The session timeout value in minutes that is assigned to the user.

The Role Assignment section displays the following information:

**Assign**

The check box for assigning a role.

**Role Name**

The name of the role for this user.

**Permissions**

The name of type of permissions for this user.

**Access Collections**

The check box for specifying access collections.

***Roles pane***

You can display administration role information for your enterprise in the Data Management Portal using the **Roles** pane. This function is available to users logged in as administrator or users with a role that has administrator permission.

The **Roles** pane contains the following information:

**Roles**

The name of the role (for example, **administrator** or **operator**).

**Application Name**

The name of the application.

**Permissions**

The type of permissions included by the role.

**Create Role**

Creates a new role and adds it to the table.

**Edit**

Edits the permissions included by a role.

**Delete**

Deletes a role and removes it from the table.

***Create Role window***

You can create a role for the users in the Data Management Portal using the **Administration** function. This function is available to users logged in as administrator or users with a role that has administrator permission.

**Role Name**

The name of the role that you are creating.

The **Create Role** window contains a Permissions section with the following information:

**Check box**

The check box corresponding to the role type and application that you are creating.

**Type**

The type of role. For example, Read, Update, Discover, or Admin.

**Application**

The application defined for this role.

**OK**

Creates the role after you enter the information.

**Cancel**

Returns to the Roles pane.

***Edit Role window***

You can edit an existing role in the Data Management Portal using the Administration function. This function is available to users logged in as administrator or users with a role that has administrator permission.

The predefined roles (administrator, operator, and supervisor) cannot be edited.

The **Create Role** window contains a Permissions section with the following information:

**Check box**

The check box corresponding to the role type and application that you are creating.

**Type**

The type of role. For example, Read, Update, Discover, or Admin.

**Application**

The application defined for this role.

**OK**

Applies changes to the role after you specify the information.

**Cancel**

Returns to the Roles pane without making any changes.

***TADDM Servers Summary pane***

The **TADDM Servers Summary** pane in the Data Management Portal (running on a streaming server deployment) contains information about discovery servers and storage servers in your environment. From this pane, you can perform various operations on the discovery servers, primary storage server, and secondary storage servers.

This pane is available only in the Data Management Portal running on a streaming server deployment.

The **TADDM Servers Summary** pane contains the following buttons:

**Refresh**

Refreshes the list of discovery servers and storage servers.

**Launch**

Connects to the selected discovery server or storage server. You can connect to the discovery server or storage server in either secure or non-secure mode.

The **TADDM Servers Summary** pane contains a table with the following fields:

**Host Name**

Displays the host name of the discovery server or storage server.

**Type**

Displays the discovery server and storage server type.

**Status**

Displays the status of the discovery server and storage server.

**Storage Pool Member**

Displays whether the storage server is used for handling discovery workload.

**Build Number**

Displays the ID of the build running on the discovery server and storage server.

**Create Component wizard**

Using the **Create Component** wizard, you can create a component.

Depending on the specific details of the component you are creating, some or all of the following pages are displayed in the **Create Component** wizard:

- General Information
- Server Information
- IP Information
- Administrative Information
- Extended Attributes

The General Information page contains some, or all, of the following information:

**Name**

Type the name of the component.

**Type**

Select the type of the component.

The Server Information page contains some, or all, of the following information:

**Available**

Lists the available content. Content can be selected from existing server type lists.

**Included**

Lists the included content.

**Add**

Adds the selected item to the **Included** list, and removes it from the **Available** list.

**Remove**

Removes the selected item from the **Included** list, and adds it to the **Available** list.

The IP Information page contains some, or all, of the following information:

**Host name**

Type the host name of the computer you want to add.

**IP address**

Type the IP address of the computer you want to add. If appropriate, move the slider to specify the subnet mask.

**Add**

Adds the specified hostname and IP address information to the list of IP addresses.

**Remove**

Removes the selected IP address from the list.

**IP Address**

The IP address of the computer.

**Subnet Mask**

The subnet mask of the computer.

**Host Name**

The host name of the computer.

The Administrative Information page contains some, or all, of the following information:

**Admin contact**

Type an administration contact for the component.

**Escalation contact**

Type an escalation contact for the component.

**Tracking number**

Type a tracking number for the component.

**Site**

Type site information for the component.

**Group name**

Type a group name for the component.

**Notes**

Type notes for the component.

The Extended Attributes page contains a field for each defined extended attribute. Extended attributes are grouped by a category. Each category has a separate tab. For each attribute, type the value of the extended attribute.

**Define Extended Attributes window**

You can create, view, and delete extended attributes.

The **Define Extended Attributes** window displays the following information:

**Component Type**

Select the component type for which you want to create or view an extended attribute.

**New**

Creates a new extended attribute for the selected component type.

**Delete**

Deletes the selected extended attribute for the selected component type.

**Extended Attribute Name**

Displays the name of the extended attribute.

**Extended Attribute Type**

Displays the type of the extended attribute.

**Category**

Displays the category of the extended attribute.

**Inherited Extended Attribute Name**

Displays the name of the inherited extended attribute.

**Inherited From Class**

Displays the class from which the extended attribute is inherited.

**Create New Extended Attribute window**

You can create extended attributes.

The **Create New Extended Attribute** window displays the following information:

**Extended attribute name**

Type the name of the extended attribute.

**Extended attribute type**

Select the type of the extended attribute. The following values are available:

- String
- Character
- Double precision floating point
- Floating point
- Integer
- Boolean
- Short integer
- Long integer

**Extended attribute category**

Select the category of the extended attribute. You can select an existing category or the category of type new.

**Domain Summary pane**

The **Domain Summary** pane in the Data Management Portal (running on the synchronization server) contains information on the domains in your environment. From this pane, you can perform various operations on the domains.

This pane is available only in the Data Management Portal running on a synchronization server.

The Distributed Domain Summary section contains the following buttons:

**New**

Adds a domain to your enterprise.

**Edit**

Edits the selected domain in your enterprise.

**Delete**

Deletes the selected domain from your enterprise.

**Refresh**

Updates the Domain Summary table information for the selected domain.

**Start**

Starts a Discovery Management Console for a domain in your enterprise.

**Start in Secure Mode**

Starts a Discovery Management Console for a domain in your enterprise using a secure SSL connection.

**SSL Connection Settings**

Displays the SSL connection settings.

The **Domain Summary** pane contains a table with the following fields:

**Domain**

Name of this domain.

**Host Name**

Name of the host for this domain.

**Last Synchronized**

The time of the last synchronization for this domain.

**Domain Status**

Status of the host.

**Add Domain and Edit Domain panes**

You can use the **Add Domain** and **Edit Domain** pane in the Data Management Portal, running on a synchronization server, to work with or change the domains that make up your enterprise. You can use these panes to add a domain to your enterprise or change an existing domain.

These panes are available only in the Data Management Portal running on a synchronization server.

The Add Domain and Edit Domain panes contain the following sections:

- **Domain Details:** Use this section to enter information describing the domain that you are adding or changing.
- **Admin Details:** Use this section to enter information about the contacts for this domain.

**Important:** To add a domain or change an existing domain, you must log in to the Data Management Portal as a user that has the Admin runtime permission.

The Domain Details section of the **Add Domain** and **Edit Domain** panes contains the following fields:

**Domain Name**

(Required) The name of the domain.

**Server Address**

(Required) The fully qualified host name or IP address of the TADDM server.

**Listening Port**

(Required) The listening port of the TADDM server. Use the inter-server service registry port of the domain. To obtain the value, use the value of the `com.ibm.cdb.service.registry.interserver.port` property from the domain server in the `$COLLATION_HOME/etc/collation.properties` file. The default value is `4160`.

In the Edit Domain pane, the fields are completed with current values.

The Admin Details section of the **Add Domain** and **Edit Domain** panes contains the following fields:

**Name**

The name of the domain administrator.

**Contact**

The contact for the domain.

**Escalation Contact**

The name of the escalation contact for the domain.

**Notes**

User notes about the domain.

The **Add Domain** and the **Edit Domain** panes contain the following buttons:

**Add Domain**

(**Add Domain** pane only) Adds this domain.

**Save Changes**

(**Edit Domain** pane only) Saves the changed information.

**Apply**

(**Edit Domain** pane only) Saves the changed information and returns to the **Domain Summary** pane.

**Cancel**

Returns to the **Domain Summary** pane without saving any information.

**Synchronize pane**

Synchronizing a domain server database with the synchronization server database requires using the **Synchronize** pane in the Data Management Portal.

This pane is available only in the Data Management Portal running on a synchronization server.

The **Synchronize** pane contains the following four sections

**Domain**

Contains the names of the domains in the enterprise.

**On Demand Synchronization**

Use this section to immediately start or stop synchronization.

**Scheduled Synchronization**

Use this section to schedule synchronization of the new domain. Synchronization information that is entered in the **Schedule Period** pane is displayed in a table.

**Last Synchronization Time**

Use this section to view synchronization details. You might need to click **Refresh** to update the time of the last synchronization.

The On Demand Synchronization section contains the following check box and buttons:

**Perform full Synchronization**

Specifies whether to perform full synchronization between the synchronization server database and the domain server database.

**Start**

Starts the synchronization.

**Stop**

Stops the synchronization.

The Scheduled Synchronization section contains the following fields and buttons:

**Name**

The name of the schedule.

**Next Synchronization**

The time of the next scheduled synchronization.

**Repeat Cycle**

Specifies how often synchronization occurs. The options are daily, hourly, or weekly.



**Interval**

The number Repeat Cycle values between each synchronization. For example, a Repeat Cycle of daily and an Interval of 2 means that 2 days pass between each synchronization.

**New**

Schedules the synchronization.

**Delete**

Removes the schedule from the Scheduled Synchronization table.

The Last Synchronization Time section contains the following button:

**View Synch Details**

Displays the synchronization status (SyncStatus) and a synchronization log.

**Schedule Period pane**

If you want to schedule synchronization, rather than performing on-demand synchronization, use the **Schedule Period** pane in the Data Management Portal (running on the synchronization server) to specify the schedule name and how often synchronization should occur.

This pane is available only in the Data Management Portal running on a synchronization server.

The **Schedule Period** pane contains the following fields:

**Name**

(Required) The schedule name.

**Start**

(Required) The calendar icon to specify the time to start the synchronization.

**Repeat**

Specifies how often synchronization occurs. The options are none, hourly, daily, or weekly.

**Every**

Required when the Repeat field is not set to None. The number of Repeat specification values between each synchronization. For example, setting Repeat to 'daily' and Every to '2 days' means that 2 days pass between each synchronization.

**New**

Adds the new synchronization period to the Scheduled Synchronization table in the **Synchronization** pane.

**Cancel**

Closes the Schedule Period pane without adding a new synchronization period.

**Searching for a component by name or IP address**

You can use the **Search** field at the top of the Data Management Portal to quickly search for discovered components by name or IP address. You can then filter the list of matching component, and you can perform actions on components directly from the search results page.

The search function searches for components that belong to any of the following subclasses:

- ComputerSystem
- AppServer
- Service
- ITSystem

To search for a discovered component by name or IP address:

1. In the **Search** field, type the string you want to search for.

You can search for a component using any of several basic identifiers:

- Display name
- Label

- Name
- Fully qualified domain name
- Numeric IP address

The **Search** function searches these attributes for any occurrences of the string you specify. For example, if you search for `raleigh.ibm.com`, all components whose fully qualified domain names include `raleigh.ibm.com` are found.

**Note:** If you need to find a component using a different attribute, you can use a custom query (available from the **Custom Query** pane of the Analytics tab).

2. Click **Search**.

After the search completes, the results are displayed in the **Search Results** panel. If a search yields multiple results, the results are displayed in a table in the multiple results workspace. If a search yields only one result, the result is displayed in the single result workspace.

3. If the search returned more than one search result, you can use the **Filters** pane to narrow the search results to the components you are interested in.

You can narrow the results table by component type. To see only components of a particular type, click the model object type in the **Component type** list. The list shows the types of all components matching the search string.

4. To perform an action on a component directly from the **Search Results** pane, do one of the following:

- In the multiple results workspace, select a row in the table and then click one of the available actions:

**Details**

Shows detailed information about the component.

**Dependencies**

Shows dependency information for the component.

**Explore**

Opens the node-centered topology view for the component.

**Changes**

Shows the change history of the component.

**Mark for Comparison**

Adds the component to the list of components to be compared.

- In the single result workspace, click the component drop-down list, and then click one of the available actions:

**Delete**

Deletes the component.

**Explore**

Opens the node-centered topology view for the component.

**Changes**

Shows the change history of the component.

**Mark for Comparison**

Adds the component to the list of components to be compared.

After searching for a component by name or IP address, search results are displayed in the single result workspace, or in a table in the multiple results workspace.

### ***Search Results pane: Multiple results workspace***

If a search yields multiple results, the results are displayed in a table in the multiple results workspace, in the Search Results pane. To open a result in the single result workspace, double click that result in the multiple results workspace.

The multiple results workspace, in the Search Results pane, contains the following information:

**Filters list**

Narrows the search results by displaying components only of the selected type in the table in the multiple results workspace.

**Component**

The fully qualified domain name of the component.

**Component Type**

The model object type of the component.

**Search Matches**

The attribute containing the matching string, and the complete attribute value (with the matching substring highlighted).

The multiple results workspace, in the Search Results pane, contains the following buttons:

**Details**

Shows detailed information about the component.

**Dependencies**

Shows dependency information for the component.

**Explore**

Opens the node-centered topology view for the component.

**Changes**

Shows the change history of the component.

**Mark for Comparison**

Adds the component to the list of components to be compared.

**Add to cart**

Adds the selected components to the cart in the **Discovered Components** pane.

***Search Results pane: Single result workspace***

If a search yields only one result, the result is displayed in the single result workspace, in the Search Results pane. For a search that yields more than one result, double click a result in the multiple results workspace to open that result in the single result workspace.

The single result workspace, in the Search Results pane, contains the following information:

**Breadcrumb trail**

Displays your position within the search results structure. To return to the multiple results workspace, click **Search Results**.

**Component drop-down list**

Lists the actions that can be performed on the current component.

**Delete**

Deletes the component.

**Explore**

Opens the node-centered topology view for the component.

**Changes**

Shows the change history of the component.

**Mark for Comparison**

Adds the component to the list of components to be compared.

**Component information**

Displays information about the component type and the time of the most recent update.

The single result workspace, in the Search Results pane, contains the following tabs:

***Dependencies tab***

The single result workspace includes the **Dependencies** tab, which displays dependency information for the search result.

The **Dependencies** tab, in the single result workspace, contains the following information:

**Dependencies explorer**

Narrows the list of dependencies by displaying components only of the selected type.

**Component**

The fully qualified domain name of the component.

**Component Type**

The model object type of the component.

The **Dependencies** tab, in the single result workspace, contains the following buttons:

**Details**

Shows detailed information about the selected component.

**Dependencies**

Shows dependency information for the selected component.

**Explore**

Opens the node-centered topology view for the selected component.

**Changes**

Shows the change history of the selected component.

**Mark for Comparison**

Adds the selected component to the list of components to be compared.

*BIRT Reports tab*

The single result workspace includes the **BIRT Reports** tab, which displays BIRT reports relevant to the search result.

The **BIRT Reports** tab, in the single result workspace, contains the following information:

**Description list**

Lists the BIRT reports, if any, that are relevant to the search result

**Report pane**

Displays the selected report.

## Task scenarios

---

These task scenarios provide instructions for completing some of the common tasks that users perform with the IBM Tivoli Application Dependency Discovery Manager (TADDM). The data and parameters that are included in these scenarios are examples only and do not represent system defaults.

### Setting up a discovery

You can use the Discovery Management Console to set up a discovery.

In this scenario, you want to discover information about the following computer systems that are behind a firewall:

- A Windows system with a host name of windows1.
- A Windows system with a host name of windows2.
- A Linux system with a host name of linux1.

This scenario has six major steps:

1. [Setting the scope](#)
2. [Configuring the access list](#)
3. [Adding gateways](#)
4. [Adding anchors](#)
5. [Running the discovery](#)
6. [Viewing the details](#)

## Step 1: Set the scope

To set the scope of a discovery, log in as a supervisor or administrator, and complete the following steps from the Discovery Management Console or the Data Management Portal.

To set the scope of a discovery, log in as a supervisor or administrator, and complete the following steps from the Discovery Management Console or the Data Management Portal:

1. In the Functions pane, click **Discovery** > **Scope** and select **Scope Sets** tab.

The **Scope** pane is displayed.

2. Do one of the following:

- In the Scope pane of the Discovery Management Console, click **Add Set**. The **Scope Set Name** window is displayed.
- In the Scope pane of the Data Management Portal, click **New Scope Set**. The **New Scope Set** window is displayed.

3. In the **Name** field, type MyScope as the name for the new scope set.

**Note:** If you are managing multiple domains with a synchronization server, make sure each scope set name is unique within all domains managed by the same server. Using the same scope set name in more than one domain can cause problems when generating reports.

4. Click **OK**. The name MyScope is displayed in the Scope Sets list.

5. Do one of the following:

- From the list of scope sets in the Discovery Management Console, select **MyScope** and click **Add**. The **Add Scope** window is displayed.
- From the list of scope sets in the Data Management Portal, select **MyScope** and click **New**. The **New Scope** window is displayed.

6. Do one of the following:

- In the **Add Scope** window of the Discovery Management Console, complete the following steps:
  - a. From the **IP Type** list, select **Host**.
  - b. In the **Hostname** field, type windows1 (assuming that the TADDM server can look up windows1 in DNS).
  - c. Click **OK**. The new scope is displayed in the list.
- In the **New Scope** window of the Data Management Portal, complete the following steps:
  - a. From the **Address** list, select **Host**.
  - b. In the **Description/hostname** field, type windows1 (assuming that the TADDM server can look up windows1 in DNS).
  - c. Click **OK**. The new scope is displayed in the list.

7. Do one of the following:

- From the list of scope sets in the Discovery Management Console, select **MyScope** and click **Add**. The **Add Scope** window is displayed.
- From the list of scope sets in the Data Management Portal, select **MyScope** and click **New**. The **New Scope** window is displayed.

8. Do one of the following:

- In the **Add Scope** window of the Discovery Management Console, complete the following steps:
  - a. From the **IP Type** list, select **Host**.
  - b. In the **Hostname** field, type windows2 (assuming that the TADDM server can look up windows2 in DNS).
  - c. Click **OK**. The new scope is displayed in the list.
- In the **New Scope** window of the Data Management Portal, complete the following steps:

- a. From the **Address** list, select **Host**.
  - b. In the **Description/hostname** field, type windows2 (assuming that the TADDM server can look up windows2 in DNS).
  - c. Click **OK**. The new scope is displayed in the list.
9. Do one of the following:
  - From the list of scope sets in the Discovery Management Console, select **MyScope** and click **Add**. The **Add Scope** window is displayed.
  - From the list of scope sets in the Data Management Portal, select **MyScope** and click **New**. The **New Scope** window is displayed.
10. Do one of the following:
  - In the Add Scope window of the Discovery Management Console, complete the following steps:
    - a. From the **IP Type** list, select **Host**.
    - b. In the **Hostname** field, type linux1 (assuming that the TADDM server can look up linux1 in DNS).
    - c. Click **OK**. The new scope is displayed in the list.
  - In the **New Scope** window of the Data Management Portal, complete the following steps:
    - a. From the **Address** list, select **Host**.
    - b. In the **Description/hostname** field, type linux1 (assuming that the TADDM server can look up linux1 in DNS).
    - c. Click **OK**. The new scope is displayed in the list.

### ***Adding scope group***

When you set the scope of a discovery, you can add the group of scope sets and add existing scope sets to the scope group.

1. To add the group of scope sets, complete the following steps:
  - a) In the Functions pane, click **Discovery > Scope** and select **Scope Groups** tab.
  - b) To create a new empty scope group, click **Add Set**. The **Scope Group Name** window is displayed.
  - c) In the **Name** field, type MyGroup as the name for the new scope group.
  - d) Click **OK**. The name MyGroup is displayed in the Scope Groups list.
2. To add existing scope sets to the scope group, complete the following steps:
  - a) From the list of Scope Groups in the **Scope Groups** tab, select **MyGroup** and click **Add**. The **Add scope sets to Group** window is displayed.
  - b) Select scope sets that you want to add to group.
  - c) Click **Add**.

## **Step 2: Configure the access list**

After you set the scope for your discovery, you must provide access information for the computer systems that you added. This information enables communications between TADDM and those computer systems during the discovery process.

To configure the access list for your discovery, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Access List**. The Access List pane is displayed.
2. To add an entry for accessing Windows systems, click **Add**. The **Access Details** notebook is displayed.
3. On the Access Information page, complete the following information:
  - a) From the **Component Type** menu, select **Computer System (Windows)**.
  - b) In the **Name** field, type windows to identify the entry in the task list.

- c) In the **Username** field, type administrator, which is the user ID to access both Windows systems.
- d) In the **Password** and **Confirm Password** fields, type password1, which is the corresponding password for administrator.
4. Optional: On the Scope Limitations page you can choose limitation to selected scopes for this access entry. You can choose the following check boxes:
  - **Entire scope** - this is the default value for access entry, there are no limitations.
  - **Limit to selected scope sets** and **Limit to selected scope groups** - Choose scope sets or scope groups. The access list entry is only used when discovering the selected scope set or scope group.
5. Click **OK**. The new access details are added to the list.
6. To add an entry for accessing the Linux system, click **Add**. The **Access Details** notebook is displayed.
7. On the Access Information page, complete the following information:
  - a) From the **Component Type** menu, select **Computer System**.
  - b) In the **Name** field, type linux to identify the entry in the task list.
  - c) In the **Username** field, type linuxusr, which is the user ID to access the Linux system.
  - d) In the **Password** and **Confirm Password** fields, type linuxusr, which is the corresponding password for linuxusr.
8. Optional: On the Scope Limitations page you can choose limitation to selected scopes for this access entry. You can choose the following check boxes:
  - **Entire scope** - this is the default value for access entry, there are no limitations.
  - **Limit to selected scope sets** and **Limit to selected scope groups** - Choose scope sets or scope groups. The access list entry is only used when discovering the selected scope set or scope group.
9. Click **OK**. The new access details are added to the list.

### Step 3: Add gateways

This scenario includes the discovery of Windows systems. To discover information about Windows systems running in your environment, you must specify a Windows system to serve as a gateway server. This gateway server must allow SSH access from the TADDM server which might require rule changes on your firewall.

To specify a gateway server, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Anchors and Gateways**. The **Anchors and Gateways** pane is displayed.
2. From the Anchors and Gateways list, click **Add**. The **Add Anchor** window is displayed.
3. From the Type list, select **Windows Gateway**.
4. Click **Host Name** to set the gateway server by its host name.
5. In the **Host Name** field, type gateway\_server, which is the name of the gateway server to use for discovering the Windows systems that you specified when you set the scope.
6. Click **OK** to save the information and return to the **Anchors and Gateways** pane.
7. After the gateway server is added, you must provide access credentials for it. To add access credentials for the gateway\_server, click **Discovery > Access List** and complete the following steps:
  - a) From the **Component Type** menu, select **Computer System (Windows)**.
  - b) In the **Name** field, type gateway to identify the entry in the task list.
  - c) In the **Username** field, type administrator, which is the user ID to access gateway\_server.
  - d) In the **Password** and **Confirm Password** fields, type gatewaypass, which is the corresponding password for administrator.
  - e) Click **OK**. The new access details are added to the list.

## Step 4: Add anchors

In this scenario, a firewall exists between the TADDM server and another section of your network, so you must enable discoveries through firewalls. To do this, you must specify at least one computer system in each section of the network that is behind a firewall. This computer system is known as an anchor.

To specify an anchor, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Anchors and Gateways**. The **Anchors and Gateways** pane is displayed.
2. From the Anchors and Gateways list, click **Add**. The **Add Anchor** window is displayed.
3. From the Type list, select **Anchor**.
4. To set the anchor by its host name, click **Host Name**.
5. In the **Host Name** field, type `anchor_server`, which is the name of the anchor to use for discovering the Linux system that you specified when you set the scope.
6. Click **OK** to save the information and to return to the **Anchors and Gateways** pane.
7. After the anchor is added, you must include it in your scope set. To add the anchor to your scope set, complete the following steps:
  - a) In the Functions pane, click **Discovery > Scope**. The **Scope** pane is displayed.
  - b) In the Scope Sets list, select **MyScope** and then click **Add**. The **Add Scope** window is displayed.
  - c) From the IP Type list, select **Host**.
  - d) In the **Hostname** field, type `anchor_server`.
  - e) Click **OK**. The new scope is displayed in the list.
8. After you add the anchor to your scope set, you must provide access credentials for it. To add access credentials for `anchor_server`, click **Discovery > Access List** and complete the following steps:
  - a) From the **Component Type** menu, select **Computer System**. If the anchor server is a Windows server, then select **Computer System (Windows)**.
  - b) To identify the entry in the task list, type `anchor` in the **Name** field.
  - c) In the **Username** field, type `ancrustr`, which is the user ID to access `anchor_server`. If the anchor server is a Windows server, then the account must have administrator privileges.
  - d) In the **Password** and **Confirm Password** fields, type `anchorpass`, which is the corresponding password for `ancrustr`.
  - e) Click **OK**. The new access details are added to the list.
9. Click **OK** to save the information and to return to the **Anchors and Gateways** pane.

## Step 5: Run the discovery

You are ready to run your discovery.

To run the discovery, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Overview**. The **Overview** pane is displayed.
2. In the **Overview** pane, click **Run Discovery**. The **Run Discovery** window is displayed.
3. From the Scope pulldown, select **Selected Scope Elements**. A tree of scopes is displayed.
4. From the tree, select **MyScope**, which is the scope set that you specified for the discovery.
5. From the Profile list, select **Level 2 Discovery**. In addition to discovering computer systems, a Level 2 Discovery also discovers applications that match custom server templates.
6. Click **OK** to run the discovery. The discovery process might take a few minutes. You can monitor the progress of your discovery in the Overview pane.

After the discovery finishes, you can view information about discovered components in the **Discovered Components** pane of the Data Management Portal.



## Step 6: View the details

You can now view all of the details about the computer systems that were discovered.

To view the details of discovered systems, complete the following steps in the Data Management Portal:

1. In the **Discovered Components** pane, complete one of the following steps:
  - Navigate to the discovered computer systems you want to view, for example **Inventory Summary > Computer Systems > Windows > windows1**
  - In the **Filter** field, type some or all of the name of the computer system you want to view, for example windows1.
2. To view the details for windows1, double-click its entry in the **Discovered Components** pane. The properties of the computer system are displayed in the Details pane. The General page is displayed by default.
3. To view other details for windows1, click the tabs located at the top of the Details pane.
4. You can perform selected actions on a computer system from the **Discovered Components** pane. To perform an action on windows1, select the check box beside the computer system name, and click **Actions > action\_name**.

For more information about the flexible approach to discovery, see the document *TADDM's Flexible Approach to Discovery* at <http://www.ibm.com/software/brandcatalog/ismlibrary/>.

## Extending custom servers

This section includes an example scenario for extending a custom server.

### Step 1: Create a custom server template

You can create a custom server template to extend a custom server.

To create a custom server template, do the following tasks:

#### *Define the general server information*

The first step in creating a custom server template is to define the general server information.

To define the general server information, do the following steps from the Discovery Management Console:

1. In the Functions pane, click **Discovery > Custom Servers**.  
The **Custom Servers** pane is displayed.
2. To add a new custom server template, click **New**.  
The **Custom Server Details** notebook is displayed.
3. In the **Name** field, type myServer1, which is the name of the new custom server that you are adding.
4. From the **Type** list, select **AppServer**, which is the type of server that you are adding.
5. To discover the myServer AppServer, click **Discover**.
6. To enable the custom server definition, click **Enabled**.

You must specify the identifying criteria for myServer1.

#### *Specify the identifying criteria*

The second step in creating a custom server template is to specify the identifying criteria.

You must define the general server information for the custom server before you specify the identifying criteria.

Identifying criteria are rules that are used to classify a discovered generic process into a known process. During discovery, the following attributes can be evaluated to determine if a CST matches:

- program name
- Windows service name

- arguments
- environment variables
- all associated ports

To specify the identifying criteria, do the following steps from the **General Info & Criteria** page of the **Custom Server Details** notebook:

1. In the Identifying Criteria section, click **All Criteria**.

This step ensures that all of the criteria matches for the process to match this CST.

2. From the first list in the Identifying Criteria section, select **Program Name**.
3. From the second list in the Identifying Criteria section, select **ends-with**.
4. In the field beside the second menu, type `java`.
5. To add another argument to the list of identifying criteria, click **Add Criterion**.

A new row is added to the Identifying Criteria section.

6. From the first list in the Identifying Criteria section, select **Port**.
7. From the second list in the Identifying Criteria section, select **is**.
8. In the field beside the second list, type `1098`.

This step is important because you know that the process uses port 1098.

**Optional:** After you specify all of the identifying criteria, click the **Config Files** tab to open **Config Files** page and add configuration files to the custom server details.

### ***Optional: Add configuration files***

After you complete the **General Info & Criteria** page, you can add configuration files for the custom server. This step is optional, however by doing these steps, a configuration file is collected from the system whenever the CST matches.

To add a configuration file, do the following steps on the **Config Files** page of the **Custom Server Details** notebook:

1. Click **Add** to add a configuration file.

The **Edit Capture File** window is displayed.

2. From the **Type** list, select **Config File**. This step specifies that the file type you are adding is a configuration file.
3. From the **Search Path** list, select **"/"** to specify that the configuration file is located in the root directory.
4. In the **Search Path** field, type `/configfiles/myServer.conf`. This step specifies the location of the configuration file.
5. Click **Capture file contents**.
6. Click **Limit size of captured file to**.
7. In the **Bytes** field, type `10000` to limit the size of the captured file to 10,000 bytes.
8. To save the configuration file information and close the **Edit Capture File** window, click **OK**.

The new configuration file is displayed on the **Config Files** page of the **Custom Server Details** notebook.

9. To save the criteria for the custom server and return to the **Custom Servers** pane, click **OK**. The `myServer` AppServer is displayed in the list of custom servers.

## **Step 2: Create a directive file**

After you complete the **Custom Server Details** notebook, you can create a directive file. A directive file contains commands and scripts that capture additional attributes that were not previously discovered. Creating a directive file is optional, but it is a necessary step in this scenario because you need to collect the `productVersion` for `myServer`.

To create a directive file, complete the following steps:

1. Open a text editor and type the following command:

```
CMD:productVersion=cat /configfiles/myServer.conf|grep"^version"|awk'{print $2}'
```

2. Save the file to the \$COLLATION\_HOME/etc/templates/commands/myServer directory.

### Step 3: Run discovery

Now you must run a discovery. This step is required because the custom servers must be discovered for the template that you created.

See [“Setting up a discovery” on page 168](#) for complete instructions on how to run a discovery.

## Business Applications

---

A business application is a collection of components that provides a business functionality that you can use internally, externally, or with other business applications. You can create business applications of individual components, which are related to each other.

For example, Order Management, Inventory Management, and Billing are business applications that might use individual components such as a Java EE application server, LDAP, and a database that runs on the Solaris server.

Business application is a type of a custom collection. You can also create the following types of custom collections:

- Collection, which is a group of any resources that you can select according to your needs.
- Access collection, which is a collection that is used to control the access to configuration items (CIs) and permissions to modify configuration items. You can create access collections only when data-level security is enabled. For more information, see the *Permissions* topic in the *TADDM Administrator's Guide*.

The following methods are provided for creating business applications:

- By using grouping patterns in Data Management Portal.
- By using application descriptors.
- By using grouping patterns that are created with Java API and loaded by the bulk load program.

### Business applications in the previous TADDM releases

In the previous TADDM versions, a business application was a flat collection of unconnected CIs. These CIs could be only top-level elements of Common Data Model. They were grouped into functional groups of elements of the same type. Each CI had to be explicitly added to a business application, either as an instance, or by using an MQL rule. It required the user to know exactly which elements formed the business application and what dependent objects the user should include into the business application.

### New approach to generate business application by using grouping patterns

With this TADDM release, the approach to creating business applications radically changes. A business application is now a graph of connected CIs of types that you specify. The most important elements while building business applications are core CIs. Core CIs provide main business value to the specific business application, for example, Java Platform, Enterprise Edition application, or a database. They are the only elements that you must add manually (either by MQL or SQL query, a set of CI instances, or as application descriptor files), or by using API (in case of integration scenarios). All other elements that constitute supporting infrastructure for a business application are added automatically by traversing relations and dependencies that are discovered and stored in TADDM. You can decide which relations are traversed and which are skipped. You can also decide which CIs from all traversed objects are used to compose the resulting business application.

As you cannot be sure that all necessary dependencies are already discovered and stored in TADDM, or because there can be dependencies that have strictly business meaning and cannot be automatically

discovered, you can specify more than one core CI for each business application. You can either create many queries to select multiple core CIs, or you can use `CustomSqlDependencyAgent` to create more dependencies between related objects. If you use the agent, you do not have to create more queries.

A grouping pattern consists of queries that select core CIs, the formula that calculates the name of business applications from the core CIs, and a description that defines the way that the dependent objects are traversed. Grouping patterns are processed automatically according to the schedule that you define and custom collections are generated as a result. Each time a grouping pattern is processed, all queries are processed, all dependencies are traversed, and the business application structure is generated according to the existing CIs and relations. As a result, all environmental changes to the structure of the business application are automatically captured and reflected when the grouping pattern is processed. For example, if a new application server, or a new virtual machine is discovered, it is added automatically to the business application. If a CI was modified, or removed, for example a virtual computer system was moved to another hypervisor, the changes are automatically applied, that is the virtual computer system is removed from the application that contained the hypervisor.

In the previous releases, one application template could produce only one business application. Now you can create not only one business application from one grouping pattern, but you can also create many business applications from one pattern, or a small set of patterns. As a result, it is easy to generalize grouping patterns to produce instances of business applications for multiple environments, for example, the application deployed in production, test, quality assurance, and performance environments. To achieve it, a grouping name expression was created. You can use it to provide a formula to calculate the name of a business application from its core CI. For example, you can use naming conventions to extract specific parts of CIs names, or any existing attributes that denote the purpose of a specific environment. You can also extend this generalization. For example, you can create a grouping pattern that generates all business applications of a given type, such as Java Platform or Enterprise Edition applications, in all deployment environments.

For more information about grouping patterns configuration and controlling grouping patterns processing, see [“Processing of grouping patterns” on page 200](#).

**Note:** Business applications that are created with the default configuration contain only high-level and middle-level objects. Therefore, some CI types, which were high-level objects in TADDM 7.2.2, are no longer high-level objects in version 7.3.0. As a result, they are not added to business applications. The new high-level objects are `SComputerSystem`, `SSoftwareServer`, `SLogicalGroup`, `SPhysicalFile`, `SSoftwareInstallation`, `SFunction`. The new middle-level object is `SDeployableComponent`. Additionally, in the new model, there is no `OperatingSystem` type, its attributes were merged into the `simple.SComputerSystem` class.

## Getting started with business applications

Before you create your business applications, you must carefully plan it. These general guidelines can help you decide how to build your business applications.

### Selecting core CIs

Core CIs are the starting point of your business application. You must first decide what is the function of your business application, and then select the appropriate core CI. An application often consists of elements that constitute the application itself, and of infrastructure on which it is deployed. If you are creating a business application for middleware, for example, a database, or an application server, it is advised to select the highest deployable object, not the elements of infrastructure. The infrastructure is what you want to discover. The object is the highest when it is on the top level of hierarchy and no other objects depend on it. For example, you can select a Java Platform, Enterprise Edition application, which is deployed on a WebSphere server. The highest deployable object is a good start in most cases.

In some cases, you may select a computer system as the core CI, for example, if the computer system does not host any software components. Otherwise, you must be very cautious in such choices because such selection generates very large applications that might be impossible to view and that might cause the TADDM server to fail with an out of memory condition.

If you are creating a business application for a stand-alone server, on which no objects can be deployed, you can select this server as the core CI.

## The size of a business application

The size of your business application depends on your needs. There is no limit to the number of nodes that you can create, but keep in mind that the larger the application, the larger the topology, and the more memory is required. Too large applications have little value because you are unable to display and analyze them. The maximum size of the topology is limited by the memory. To prevent the crash of the TADDM server, a safety check is implemented. The limit depends on the current maximum Java heap size settings for Tomcat JVM process (TADDM 7.3.0) or for Liberty JVM process (TADDM 7.3.0.1, and later). The settings are defined based on a linear function  $(25 * M / 32) - 200$ , where M is the maximum Java heap size. For example, the topology that has 600 nodes, requires 1 GB heap. The topology that has 3000 nodes, requires 4 GB heap. The limit applies to all topologies that are simultaneously displayed in multiple browsers, not just to the current user.

To ensure that the topology is not too large, you might create an application, set the `maxHopsLimit` parameter to a lower value, for example 2, and analyze the resulting application. If you notice that the traversal is not deep enough, you can set this parameter to a higher value. You can analyze what relations and dependencies are added, how data is traversed. Then, you can add more elements if you think that the application lacks important components, or filter some of the elements that are meaningless to your application. To add and filter dependencies and relations, use the grouping pattern configuration.

The size of the business applications is correlated with the schedules of their execution. The interval at which the building process starts should be higher than the total time of building the application. For example, if you have a large application, which is built for a couple of hours, increase the interval to, for example, 20 hours.

## Schedules

The applications are rebuilt at the specified intervals. The default interval is every 4 hours. Consider what changes are made to your business applications, and how often, and set the interval accordingly. If an application is critical to your business, set it to run more often, for example every day, or twice a day. If an application is less important, you can set it to run less frequently, for example once a week. It is better when not all of your applications are rebuilt at the same time. However, if you are sure that you have enough threads to process all of your grouping patterns, you can schedule them to run at the same time. If you plan your schedules carefully, you can avoid a situation in which there are no available threads to process grouping patterns, which might cause a critical application to wait for being updated.

Use the `bizappscli` tool to configure the schedules of your applications.

Adjust the schedule to the size of the business application as specified in the previous section.

## Related reference

[“Configuring the maximum size of displayed topologies” on page 199](#)

Displaying large topologies is memory consuming and might even lead to a crash of the TADDM server. To prevent that, a safety check is implemented. You can also configure the maximum size of the topology, depending on your needs, in the `collation.properties` file.

[“Grouping patterns configuration” on page 200](#)

You can control the process of business application creation by using grouping patterns configuration. The configuration allows you to include or exclude chosen relations during data traversal and chosen classes from resulting business applications. You can also change the dependency direction for relations that are defined in Common Data Model, and assign tiers to business application elements.

[“Actions for managing grouping pattern schedules” on page 222](#)

By using the `bizappscli` tool, you can create and modify grouping patterns schedules.

## Business application structure

Business application structure is created automatically, basing on a grouping pattern definition and on grouping pattern selectors definitions.

For details, see [“Creating grouping patterns” on page 179](#).

The structure of business applications has a form of a directed graph. The graph elements are called Nodes and the graph edges are called Paths.

## Business application

Business application is represented by a new class in Common Data Model named

```
com.collation.platform.model.topology.customCollection.CustomCollection
```

CustomCollection has a broader meaning and can represent not only business applications but also other types of collections that do not have strictly business meaning. The `hierarchyType` attribute defines the meaning for each custom collection. Currently, this attribute can have three values: Business Application, Collection, and Access Collection. Access Collection is used by TADDM to control access to some sets of data. For more information about data-level security, see the *Permissions* topic in the *TADDM Administrator's Guide*.

## Business application nodes

A business application node always points to only one component, which usually is a configuration item. Node is represented by a new class in Common Data Model named `com.collation.platform.model.topology.customCollection.Node`.

A node is not visible in Data Management Portal. It means that when you display the details pane for a particular node, the details pane of a component with which the node is connected is displayed.

For these types of elements which are less important or irrelevant for the overall structure, nodes are not created. Such elements are called low-level elements, and typically they are not configuration items, for example IP address, operating system, process, CPU.

## Business application paths

A business application path is a connection between two business application nodes. Path is represented by a new class in Common Data Model named `com.collation.platform.model.topology.customCollection.Path`. The path is always directed according to objects' dependency direction. It means that the dependent object points to the object on which it depends.

A path contains information about the source and target nodes, and about a route from a source node to a target node.

Paths represent relations or dependencies between source and target objects, but they can also represent entire subgraphs. The subgraphs include all nodes that were not included as nodes in the business application, as defined in the composition configuration of the grouping pattern that generated the specific business application. All elements that are excluded by the composition configuration, and relations between them, are stored as detailed information for just one path between a specific pair of objects. You can view these details in a path's detail pane.

Two objects can be connected by more than one route. For example, direct dependency between the source and target objects, direct relation between source and target objects, and a route that connects a couple of low-level objects through a set of relations and dependencies. However, all objects that were added during traversal are represented by just one path, and the detailed data (the XD attribute) contains additional information for each route.

For details about business application paths, see [“Business Application Path details pane” on page 144](#).

The CustomCollection class does not have relations to nodes and paths, but the Node class and the Path class have relations to the CustomCollection class. If you want to query all nodes or paths for a specific custom collection, query those nodes or paths, the parent attribute of which points to a specific custom collection. For example:

```
select * from Path where parent.guid == 'DED47778C834ABAFBA6A55137D1A8B'
```

## Querying routes

Each CI can have additional data that is stored as XML along with the object. The attribute that contains XML data is named XD. In case of a path object, this attribute stores detailed information about low-level objects that were traversed during business application generation but were excluded by the composition configuration. It also stores detailed information about these objects. For details about composition configuration, see [“Composition configuration” on page 205](#).

The following example shows an output from querying a path object:

```

<Path array="1" guid="F92431E223E637ECAC3775DD54AA0AC2"
  lastModified="1413539533336"
  parent="DED47778CBA834ABAFBA6A55137D1A8B" xsi:type="coll:com.collation.
platform.model.topology.customCollection.Path">
  <sourceNodeGuid>5B2C7E013F4A3E948080405446FC38DD</sourceNodeGuid>
  <targetNodeGuid>24EEC97B70F43039AE2EC88C31D96B56</targetNodeGuid>
  <displayName>1.0.0.0/24 - 1.0.0.7</displayName>
  <bidiflag>3</bidiflag>
  <XD>
    <xml>
      <routes>
        <instance>
          <routeStart/>
          <fromObjectGuid>1AA04147456D3FB3811DDC1425732B56</
fromObjectGuid>
            <relationshipType>net.Interface(ipNetwork) -&gt;
relation.Networks -&gt; net.Interface</relationshipType>

            <toObjectGuid>EB377A0AB0BA35E9A78688FACEFE181E</toObjectGuid>

            <fromObjectGuid>1AA04147456D3FB3811DDC1425732B56</
fromObjectGuid>
            <relationshipType>net.Interface(parent) -&gt;
relation.Contains -&gt; sys.ComputerSystem(ipInterfaces)</relationshipType>

            <toObjectGuid>4FF04D906FD8354298DCB2F9116AD0C3</toObjectGuid>
          </instance>
        </routes>
      </xml>
    </XD>
    <isPlaceholder>false</isPlaceholder>
    <status>1</status>
    <statusChangeTime>1413539533336</statusChangeTime>
  </Path>

```

To query inside the XD attribute, you can use a new eval operator. For example, to query paths that contain a specific object as a low-level object, use the following query:

```

select * from Path where XD eval
'/xml/routes/instance[fromObjectGuid=\"1AA04147456D3FB3811DDC1425732B56\"]'

```

For more information about MQL and the eval operator, see the *Model Query Language* topic in the *TADDM SDK Developer's Guide*.

**Note:** Each route has segments. A segment consists of two nodes that are connected by a relationship or dependency. If the number of segments exceeds the limit that is defined in the `com.ibm.cdb.serviceinfrastructure.path.max.length` property, the route is not created. For details, see “Configuring the `collation.properties` file entries” on page 231.

## Creating business applications with grouping patterns

You can create business applications by defining grouping patterns in Data Management Portal.

### Creating grouping patterns

You can create new grouping patterns in the **Grouping Patterns** pane, in the Data Management Portal.

To create a new grouping pattern, complete the following steps:

1. Open Data Management Portal.
2. In the Functions pane, click **Discovery** if you use TADDM 7.3.0, or 7.3.0.1, or **Analytics**, if you use TADDM 7.3.0.2, or later.
3. Click **Grouping Patterns**. The **Grouping Patterns** table is displayed.
4. Click **New**. The **General Information** pane of the **Create new Grouping Pattern** window is displayed.
5. In the **Name** field, specify a grouping pattern name. The name can be changed to a different one and it does not have to be unique.



6. Select the pattern type and select the pattern compatibility type if applicable.
  - a) The following pattern types are available:
    - **Business Application**, which is a collection of components that provides a business functionality that you can use internally, externally, or with other business applications.
    - **Access Collection**, which is a collection that is used to control the access to configuration items (CIs) and permissions to modify them when data-level security is enabled.
    - **Collection**, which is a group of any resources that you can select according to your needs.
  - b) **Fix Pack 1**  
 You can select the pattern compatibility type only when the compatibility with earlier version mode is enabled. This type determines the type into which the old grouping entity is converted. For more information about the conversion process, see [“Migration from 7.2.2 and automatic conversion of old business applications”](#) on page 233.  
 The compatibility type is based on a chosen pattern type:
    - **Collection** - only **Collection** compatibility type is available.
    - **Access Collection** - only **Access Collection** compatibility type is available.
    - **Business Application** - the following types are available:
      - **Business Application**,
      - **Business Service**.
7. Specify the value for the **Schedule** option. This is the time when business applications are generated. You can choose from the predefined values, or change the time by using the `bizappscli` tool.
8. Specify the value for the **Configuration** option. You can choose from the predefined values, or change the configuration by using the `bizappscli` tool. The tool allows you to modify advanced options of the business application building process.
9. Optional: Enter a brief description in the **Description** area.
10. Optional: If applicable, in the **URL** field, specify the URL that links to the business application.
11. Click **Next**. The Selectors pane is displayed.  
**Note:** You can create a draft version of a selector by selecting the **Disabled** check box. Such selector is not used by BizAppsAgent in building custom collections.
12. Click **New** to add a selector, and then enter the name of the selector.
13. **Fix Pack 1**  
 Optional: In the **Tier name** field, specify the tier name of the selector. For more information, see [“Business application tiers”](#) on page 235.
14. Optional: Enter the description of the selector in the **Description** area.
15. Define selection rules. Click **Choose...**
16. Click **Choose...** to edit core CIs selection method and grouping name expression. A new window is displayed.
  - a) Choose core CI selection type. The following methods are available:
    - **MQL query** - enter a query in the text area. The starting "SELECT \* FROM" is already hardcoded, so you must specify only source table and optional conditions. For more information, see the *Model Query Language overview* topic in the *TADDM SDK Developer's Guide*.
    - **SQL query** - enter a query in the text area. The starting "SELECT \* FROM" is already hardcoded, so you must specify only source table and optional conditions. For this type of query, use the building blocks database view names rather than the table names, for example `BB_COMPUTERSYSTEM40_V WHERE NAME_C like 'test%'`.
    - **Instance-based selection** - select the CIs manually from the list of all discovered CIs.**Note:** You can define custom queries by clicking **Get from templates**. Custom queries can be saved and reused in other selectors.



- b) Edit the **Grouping name expression** field. Enter grouping name expression for the selector or leave the default value "`${patternName}`". The default value sets the name of the generated collection to the name of the grouping pattern.

For more information about grouping name expression, see [“Grouping name expression”](#) on page 195.

- c) Optional: Click **Test** to verify that the selection method and grouping name expression are valid. If there is an error, the relevant field is highlighted and the detailed error message is indicated by the tooltip. If there are no validation errors, you can see the sample results. For each core CI, a computed name of the custom collection is included. If the grouping name expression is not applicable for some of the core CIs, for example when an attribute is not set for the object, a warning is displayed. Custom collections are not created for such core CIs.

**Note:** Even if you do not test the query and grouping name expression, they are validated automatically when you save the entire grouping pattern. If there are any errors, the grouping pattern is not saved, unless the error refers to the disabled selector.

- d) Click **OK** to accept the changes and close the dialog window.

The validation status of each selector is represented by an icon on the selectors list. You can move mouse over the icon to see a tooltip message with details.

17. Select or clear options regarding Dependency Traversal Template, which affect how related CIs are added to the collection.
- Clear the **Dependency Traversal Template** option to skip traversing dependencies of the core CI. In such a case, custom collections that are generated by this selector consist only of CIs specified by the query of selector.
  - If you select the **Dependency Traversal Template** option, the following options are available:
    - **Higher Up** - traverse the dependencies up. When you select this option, BizAppsAgent traverses all objects, which depend on a specific core CI, and all other objects that depend on these objects. For example, BizAppsAgent traverses from a hypervisor to all computer systems that are hosted on it, and then to all application servers that are hosted on these computer systems.
    - **Higher Down** - traverse the dependencies up and then down. You can select this option only when the HigherUp option is selected. With the HigherDown option selected, BizAppsAgent goes down from all objects that it encounters while it traverses up the dependencies.
    - **Lower Down** - traverse the dependencies down. When you select this option, BizAppsAgent traverses all objects, on which a specific core CI depends, and all other objects on which these dependent objects depend. For example, BizAppsAgent traverses from the web server to a hosting computer system, on which the web server depends, and then goes from the computer system to a hosting hypervisor, on which the computer system depends.
    - **Lower Up** - traverse the dependencies down and then up. You can select this option only when the LowerDown option is selected. With the LowerUp option selected, BizAppsAgent goes up from all objects that it encounters while it traverses the dependencies down.
18. Optional: You can add more selectors to this grouping pattern by clicking **New** and repeating steps 12 - 17.
19. Optional: You can review other selectors of this grouping pattern by clicking a selector from the list on the left of the window. You can then change the details of each selector.
20. Optional: You can remove a selector by selecting it from the list on the left of the window and clicking **Delete**.
21. Click **Next**. The **Administrative Information** pane is displayed.
22. Optional: Specify the appropriate information in the **Administrative Information** pane.
23. Optional: Click **Next**. The **Extended Attributes** pane is displayed.

**Note:** This pane is displayed only if extended attributes for the Grouping Pattern type are defined.

To learn more about extended attributes and their definitions, see the *Extended attributes* topic in the TADDM SDK Developer's Guide.

24. Optional: Specify the appropriate information in the Extended Attributes pane. These extended attributes are copied to the generated custom collections.

**Notes:**

- There is no need to add extended attributes definitions to the custom collection type explicitly. Required definitions are automatically created by BizAppsAgent when generating custom collections.
- When extended attributes' definitions are automatically cloned from Grouping Pattern to Custom Collection type, the type of each extended attribute is widened to String to avoid data loss while casting existing values.
- Every time when grouping patterns are processed, the existing extended attribute values of a custom collection are updated. The manual changes to propagated extended attributes of a custom collection are lost on the next execution of the grouping pattern. If custom collections created from a single grouping pattern are to have an extended attribute that can be customized for each custom collection independently, then you must define a dedicated extended attribute for CustomCollection type with a unique GroupingPattern name and category.

25. Click **Finish** to close the wizard and submit the newly created grouping pattern. The **Grouping Patterns** table is refreshed automatically.

**Related reference**

[“bizappscli tool” on page 216](#)

You can use the CLI `bizappscli` tool to manage grouping patterns, grouping pattern processing schedules, grouping pattern configurations and the execution of grouping patterns.

**Fix Pack 2** You can use the tool to create reports for analyzing the content of business applications.

**Fix Pack 3** You can use the tool to export the graph of the business application topology to the SVG format.

## Displaying grouping patterns

You can display details of grouping patterns in the **Grouping Patterns** pane, in the Data Management Portal.

**Note:** **Fix Pack 3** To display the Grouping Patterns pane with its content, you must have the Update permission granted for the DefaultAccessCollection.

To display the grouping patterns information, complete the following steps:

1. In the Functions pane, click **Discovery** if you use TADDM 7.3.0, or 7.3.0.1, or **Analytics**, if you use TADDM 7.3.0.2, or later.
2. Click **Grouping Patterns**. The **Grouping Patterns** table is displayed.
3. Optional: If during processing the grouping pattern by BizAppsAgent errors occur, a warning icon is displayed. To see the detailed error message, move the mouse over the icon or click the **Edit** button to open the **General Information** tab.

**Note:** When a grouping pattern is processed and during the core CI calculations an error occurs, for example the provided query is invalid, the processing of this grouping pattern is abandoned. The custom collections content of any selector is not calculated, therefore custom collections stay unchanged. Other grouping patterns are processed normally.

- You can refresh the list of the grouping patterns that are displayed in the **Grouping Patterns** table. Click **Refresh View**.
- You can sort the grouping patterns in the **Grouping Patterns** table. For example, to sort the grouping patterns by type, click the **Type** column.

## Editing grouping patterns

You can edit an existing grouping pattern in the **Grouping Patterns** pane, in the Data Management Portal.

To edit an existing grouping pattern, complete the following steps:

1. In the Functions pane, click **Discovery** if you use TADDM 7.3.0, or 7.3.0.1, or **Analytics**, if you use TADDM 7.3.0.2, or later.
2. Click **Grouping Patterns**. The **Grouping Patterns** table is displayed.
3. Select the pattern that you want to edit and click **Edit**.
4. Edit one or more pattern attributes.

To learn about the UI for editing attributes of a grouping pattern and its selectors, see [“Creating grouping patterns” on page 179](#). To modify selectors of the grouping pattern, go to the **Selectors** tab.

5. To save the changes, click **OK**.

## Deleting grouping patterns

You can delete an existing grouping pattern in the **Grouping Patterns** pane, in the Data Management Portal.

To delete an existing grouping pattern, complete the following steps:

1. In the Functions pane, click **Discovery** if you use TADDM 7.3.0, or 7.3.0.1, or **Analytics**, if you use TADDM 7.3.0.2, or later.
2. Click **Grouping Patterns**. The **Grouping Patterns** table is displayed.
3. Select the pattern that you want to delete and click **Delete**.
4. To save the changes, click **OK**.

The selected grouping pattern is removed.

## Creating business applications with application descriptors

Application descriptors in earlier TADDM versions were used to help automate the process of discovering, creating, and maintaining business applications. Now, they are an important part of a new mechanism of Business Application Composition. New grouping patterns are cyclically created and the existing grouping patterns are cyclically modified by using data from discovered application descriptors.

Using application descriptors on computer systems and software servers allows for automatic inclusion of these computer systems, servers, along with other components deployed on them and software modules to a specific business application. You can add such system and its components to a business application just by putting an appropriate application descriptor XML file in the predefined location or in the location specified in the `collation.properties` file. You do not need to use TADDM UI or any configuration. Application descriptor XML files are read during a regular TADDM infrastructure discovery. After this process, the application descriptor agent, together with business application agent, periodically processes all discovered descriptors and generates appropriate selectors, or grouping patterns if necessary, from these discovered descriptors. The generated selectors or grouping patterns are then processed by the business application agent, which generates target business applications.

Selectors are generated together by `AppDescriptorAgent` and `BizAppsAgent`.

`AppDescriptorAgent` only creates new grouping patterns from new application descriptors. It is the only task that it executes. It does not refresh selectors based on application descriptors for the existing grouping patterns. `BizAppsAgent` refreshes such selectors for all business application names that were generated from a specific grouping pattern. To make sure that the grouping patterns contain up-to-date data, when they are created, they are cyclically processed by `BizAppsAgent`. During the processing of a specific grouping pattern, `BizAppsAgent` first processes selector types other than the one based on application descriptors. When the agent finishes creating all business applications, it collects their names and refreshes selectors based on application descriptors. For example, the agent creates new selectors, modifies existing selectors or removes a selector if the application descriptor was removed.

To create grouping patterns and selectors based on application descriptors, all application descriptors that are stored in the TADDM database are read and grouped by the name of the target business application. For each business application, a list of components and functional groups that are assigned to them is created by using CIs that are provided in component application descriptors. This list is then divided by the names of functional groups, and selectors specific to each functional group are created.

The functional group name is used as a tier name on the generated selector. For more information about tiers configuration, see [“Tiers configuration” on page 207](#).

If the base application descriptor provides the name of the grouping pattern, this grouping pattern is updated with the new list of selectors or a new grouping pattern is created. If the grouping pattern is not provided, the agent checks whether the target business application exists. If it exists, the agent finds the grouping pattern that generated this business application (it can be either grouping pattern for one business application, or more general grouping pattern that generates a group of business applications) and attaches the generated selectors to this grouping pattern. If the target business application does not exist, a new grouping pattern is created to generate just this one business application. If a specific component application descriptor was removed from the target machine and this change was reflected in the TADDM database after the discovery, the BizAppsAgent removes these components from existing selectors. If, after this modification, selector becomes empty, it is also removed. If a grouping pattern no longer contains any selectors, it is disabled and, as a result, removed from processing.

**Note:** When you define a grouping pattern for many component application descriptors, it must be the same for the base application descriptor and associated component application descriptors. If the associated component application descriptors define various grouping patterns, no pattern is created, even if the base application descriptor does not define any grouping pattern.

You can view the selectors and grouping patterns generated by the application descriptor by using TADDM UI. However, since the selectors of the application descriptor type are generated automatically, you cannot modify or delete them. You can modify only dependency traversal template for them.

In the current release, application descriptors are stored in a new class named BizAppDescriptor. During migration, all existing application descriptors are transformed to this new class. Grouping patterns that are generated during migration use these migrated application descriptors.

You can use the following strategies for creating and deploying application descriptors:

- Define application during development and deployment. By doing so, you can capture the most accurate and complete information about the packaging of modules in business applications.
- After the application is already in place (by creating the descriptors and putting them in the appropriate location on the target computer systems), you can deploy application descriptors.

## Step 1: Creating the base application descriptor

The first step in creating the Finance application is to create the base application descriptor.

The base application descriptor contains general information about the application, such as the version and contact information. Only one base application descriptor is required for the Finance application. If more than one descriptor is deployed, then the one with the most recent time stamp is processed by TADDM.

**Important:** You do not have to use a specific naming convention for XML files. When you use .xml extensions and deploy the files in the appropriate application descriptor directory, they are processed by TADDM.

In the previous version of TADDM, base application descriptor allowed for specifying the application definition inside the base application descriptors. This is no longer supported. All application definition sections are now ignored during the application descriptor agent processing. The appropriate warning message is printed in the logs.

**Note:** The base application descriptor defines only general information about the business application. This information is stored in a grouping pattern which is used to create a target business application. However, grouping patterns and business applications must have the content, and therefore, base application descriptors must be accompanied by component application descriptors. A base application descriptor alone cannot create a grouping pattern or a business application.

To create the base application descriptor for the Finance application, complete the following steps:

1. Using a text editor, open the `finance_base_app_desc.xml` file.
2. In the .xml file, specify the following schema:

```
<base-app-descriptor>
  <app-instance
    name="Finance"
    grouping-pattern="Finance1"
    version="1.0"
    description="All components of the finance application"
    url="http://finance.acme.com"
    contact="John Doe" />
</base-app-descriptor>
```

You must specify a value only for the `app-instance name` field. All other fields are optional.

3. Save the `.xml` file.

## Step 2: Creating the component application descriptors

After you create the base application descriptor, you must create component application descriptors.

The component application descriptor contains information about a specific computer system or server and software component or module deployed within a server, along with information about the participation of the entity in the business application. Entities can include computer systems, database servers, Java EE servers, modules within a server such as Web applications, Enterprise Archives, and JSP pages or other components deployed on the servers.

The component application descriptors have the following schema:

```
<component-app-descriptor
  app-instance-name="Finance"
  grouping-pattern="Finance1">
  <component-descriptor
    type="server"
    name="htdocs"
    functional-group="deprecated"
    marker-module="true" />
</component-app-descriptor>
```

**Note:** The `grouping-pattern` attribute is optional.

The `component-descriptor` has the following attributes:

### type

You can specify four type values:

- `host` - used when you add computer systems to a business application.
- `server` - used when you add software servers or software application servers to a business application.
- `module` - used when you add software modules deployed on the server to a business application.
- `deployable` - used when you add other, general components deployed on the server to a business application.

### name

Used to specify the software module or deployed component that is selected to a business application. The value of this attribute depends on the value of the `marker-module` flag.

**Note:** For computer systems and software servers this attribute is not used.

### functional-group

Deprecated. This attribute is supported only to provide compatibility with old application descriptors. A special tier (TADDM 7.3.0) or a manual tier (TADDM 7.3.0.1, and later) is created with a name equal to this value. For more information about special and manual tiers, see the *Business entities compatibility with earlier versions* topic in the *TADDM Administrator's Guide*.

It is not advised to use this attribute, but to rely on regular tiers rules. For more information, see [“Business application tiers” on page 235](#).

### marker-module

A boolean flag that indicates whether the selected software module or deployed component is a "marker" or not. The allowed values are `true` or `false`.

- If a software module is used as a "marker", all software modules deployed on the server or server domain are added to the business application via the grouping pattern selector when a module with that name is encountered. If the server is a WebSphere deployment manager server, all software modules deployed on all servers in all nodes in the WebSphere cell managed by this deployment manager are added to the business application via the grouping pattern selector. Similarly, if the server is the WebLogic administration server, all software modules from the domain managed by this server are added to the business application via the grouping pattern selector.
- If a deployed component denoted by type="deployable" is used as a "marker", all deployed components on the server are added to the business application via the grouping pattern selector.

There can be multiple component-descriptor sections in a single component application descriptor. As a result, you can define multiple elements to be added to the business application without creating multiple files.

The following components are included in the Finance business application that is provided as an example:

- The Apache web server named "apache\_server1".
- The IBM WebSphere named server "j2ee\_websphere1".
- The DB2 instance named "my\_db2".

As those components are on different physical machines, a component application descriptor must be created for each component of the Finance application.

### ***Step 2a: Creating the descriptor for the Apache server***

Complete the following steps to create the component application descriptor for the Apache server.

Complete the following steps to create the component application server for the Apache server:

1. Open a text editor and create the apache\_coll\_desc.xml file.
2. In the apache\_coll\_desc.xml file, specify the following schema:

```
<component-app-descriptor
  app-instance-name="Finance">
  <component-descriptor
    type="module"
    name="mod_ldap"
    marker-module="false" />
  </component-app-descriptor>
```

3. Save the apache\_coll\_desc.xml file in the apache\_server\_root/appdescriptors directory on the Apache server.
4. Restart the Apache server.

### ***Step 2b: Creating the descriptor for the IBM WebSphere server***

Complete the following steps to create the component application descriptor for the IBM WebSphere server.

Complete the following steps to create the component application server for the IBM WebSphere server:

1. Open a text editor and create the websphere\_coll\_desc.xml file.
2. In the websphere\_coll\_desc.xml file, specify the following schema:

```
<component-app-descriptor
  app-instance-name="Finance">
  <component-descriptor
    type="module"
    name="FinanceEJB"
    marker-module="false" />
  </component-app-descriptor>
```

3. Save the websphere\_coll\_desc.xml file in the WebSphere\_profile\_dir/appdescriptors directory on the WebSphere server.
4. Restart the WebSphere server.

### Step 2c: Creating the descriptor for the DB2 instance

Complete the following steps to create the component application descriptor for the DB2 instance.

Complete the following steps to create the component application descriptor for the DB2 instance:

1. Open a text editor and create the `db2_coll_desc.xml` file.
2. In the `db2_coll_desc.xml` file, specify the following schema:

```
<component-app-descriptor
  app-instance-name="Finance">
  <component-descriptor
    type="module"
    name="Finance"
    marker-module="false" />
  </component-app-descriptor>
```

3. Save the `db2_coll_desc.xml` file in the `$DB2INSTANCEHOME/appdescriptors` directory on the computer system where the DB2 instance is located.

### Step 3: Running a discovery

After you complete the application descriptors, you must run a discovery. This step is required because all components of the Finance application must be discovered before you can organize them into a business application.

See [“Setting up a discovery” on page 168](#) for complete instructions on how to run a discovery.

**Restriction:** Application descriptors used for business applications are not supported by the script-based or asynchronous discovery sensors. During the script-based or asynchronous discovery, the application descriptors are not discovered.

### Step 4: Viewing the business application

After you complete the application descriptors and run a discovery, you can view details about the Finance business application.

To view details about the Finance business application, complete the following steps from the Discovery Management Console:

1. In the Functions pane, click **Topology > Finance**.

The **Business Applications - Finance** pane is displayed. The **Business Applications - Finance** pane contains the following components:

- **apache\_server1**
- **j2ee\_websphere1**
- **my\_db2**

2. In the **Business Applications - Finance** pane, select **apache\_server1**.

Details about **apache\_server1** are displayed in the **Details** pane.

3. To view details about either the **j2ee\_websphere1** or **my\_db2** components, select the appropriate icon in the **Business Applications - Finance** pane. Details about the selected component are displayed in the **Details** pane.

#### Related reference

[“Configuring the maximum size of displayed topologies” on page 199](#)

Displaying large topologies is memory consuming and might even lead to a crash of the TADDM server. To prevent that, a safety check is implemented. You can also configure the maximum size of the topology, depending on your needs, in the `collation.properties` file.

## Creating business applications with Java API

You can create and manage business applications by using Java API. You can also load grouping patterns into the TADDM database by using the bulk load program.

## Managing grouping patterns by using Java API

Grouping pattern methods enable creation, modification, deletion, and retrieval of grouping patterns together with their selectors.

Grouping patterns define rules that are used to build business applications. Patterns are applied periodically to the TADDM database to create business applications, collections, or access collections.

Grouping patterns specify the selectors that define starting points and rules of topology traversal that is performed to create the resulting business applications, collections, or access collections. A selector defines the way of selecting configuration items from the TADDM database.

The selected configuration items become objects called 'core CIs'. For example, selectors of the MQL or SQL type contain a query that returns the list of such core CIs. By using those objects and applying relationships traversal template to them, the process of building a collection begins.

**Important:** The date and time on all TADDM servers should be synchronized.

### Grouping pattern attributes

Each grouping pattern must consist of the following attributes:

#### **name**

The name of the grouping pattern.

#### **hierarchyType**

The type of result that this pattern defines. The allowed values for hierarchy type are "BusinessApplication", "Collection" and "AccessCollection". Depending on the value of hierarchy type, the collection of appropriate type is created after topology traversal.

### Selectors attributes

Each grouping pattern specifies one or more selectors with the following attributes:

#### **name**

The name of the selector.

#### **parent**

The parent grouping pattern that contains this selector. Selectors never exist alone. They are always a part of grouping patterns.

#### **isDisabled**

An option to create a draft version of a selector. Disabled selectors are not used by BizAppsAgent in building custom collections.

#### **type**

The type of the selector. The following list contains the allowed values:

- MQL (0) - the MQL query is used to define starting point configuration items for this selector,
- SQL (1) - the SQL query is used to define starting point configuration items for this selector,
- Application descriptor (2) - the application descriptor defines the starting point configuration items for this selector,
- Manual (3) - only the manually selected configuration items are used as starting points for this selector.

#### **query**

For selectors of MQL or SQL type, this field is mandatory. It holds the appropriate query string that is used to select starting point configuration items. In case of SQL query, the user can use any syntax that is supported by the database that TADDM server is installed to. Only valid SELECT statements are acceptable during grouping pattern creation. Any other SQL statements, for example INSERT, UPDATE, DELETE, DROP, fail with the exception that denotes invalid query syntax. In case of MQL queries, there is no query validation.

#### **GroupingNameExpression**

The definition of a rule that is used to generate grouping name expression of the collections that are created by using this selector.



### **useTraversalTemplate**

A boolean flag that defines whether this selector specifies the traversal template. The traversal template defines rules of topology traversal when building collections. If the **useTraversalTemplate** flag is not set, collections elements built from this selector contain only starting point elements (core configuration items; for example configuration items that are found by using MQL or SQL queries). No other elements are traversed and added to a collection. If the **useTraversalTemplate** flag is set to `true`, rules that are defined in the fields **goHigherUp**, **goHigherDown**, **goLowerDown**, **goLowerUp** are used to define traversal pattern.

### **goHigherUp**

A boolean flag that specifies whether traversal goes through relationships and dependencies that point to the current configuration item up to the higher configuration item, which is the source of the relationship or dependency. Then, traversal continues going up through all relationships and dependencies that point to the current configuration item from the source configuration items of those dependencies or relationships.

### **goHigherDown**

A boolean flag that specifies whether traversal goes through relationships and dependencies that point from the current configuration item down to the target configuration item. This rule is used only if traversal reaches current configuration item by using at least one step of the **goHigherUp** rule.

### **goLowerDown**

A boolean flag that specifies whether traversal goes through relationships and dependencies that point from the current configuration item down to the lower configuration item which is the target of the relationship or dependency. After that traversal continues going down through all relationships and dependencies that point from the current configuration item to the target configuration item of those dependencies or relationships.

### **goLowerUp**

A boolean flag that specifies whether traversal goes through relationships and dependencies that point to the current configuration item from the upper source configuration item. This rule is used only if traversal reaches current configuration item by using at least one step of the **goLowerDown** rule.

## **Grouping patterns API methods**

The following table lists the grouping patterns methods and provides the descriptions of these methods.

Table 24. Grouping pattern API methods	
Method	Description
<code>createGroupingPattern(GroupingPattern pattern)</code>	Creates the grouping pattern with specified selectors and parameters. The sole pattern parameter contains the definition of grouping pattern to be created together with all its selectors.
<code>getAllGroupingPatterns()</code>	Retrieves all grouping patterns that exist in the TADDM database.
<code>getGroupingPattern(Guid guid)</code>	Retrieves the grouping pattern with specified guid.
<code>removeGroupingPattern(Guid guid)</code>	Deletes the grouping pattern with specified guid.
<code>updateGroupingPattern(GroupingPattern pattern)</code>	Updates the specified grouping pattern.

Grouping pattern methods provide means to control schedules creation, modification, deletion, and retrieval.

## Grouping pattern schedules

### Pattern schedule attributes

Each pattern schedule must consist of the following attributes:

#### **name**

The name of the schedule that is used to identify the schedule. While the name does not need to be unique, it is advised not to create multiple schedules with the same name.

#### **ExecutionGroupId**

The name of the execution group that this schedule is part of.

The creation of a schedule with execution group does not automatically mean that any new thread pool is created on any of the storage servers. That is separately controlled by properties on each storage server.

#### **description**

The description of the schedule.

### Pattern schedule additional attributes

In addition, each pattern schedule must consist of one of the following attributes:

#### **intervalInMinutes**

How often (in minutes) the schedule is fired.

or

#### **cronExpression**

Expression in cron format that provides enhanced control over patterns execution cycles.

**Note:** To use pattern scheduling methods via JavaAPI, a number of additional jar files must be available in the API client's class path. You can find these jar files in the TADDM\_HOME/lib directory (also referred as LIB):

- LIB/schedules.jar
- LIB/quartz/c3p0-0.9.1.1.jar
- LIB/quartz/quartz-2.2.1.jar
- LIB/quartz/quartz-jobs-2.2.1.jar

The following table lists pattern schedule management methods and provides the descriptions of those methods:

Table 25. Pattern schedule management	
Method	Description
getPatternSchedules()	Retrieves all schedules that exist in TADDM database
addSchedule(PatternSchedule schedule)	Creates new pattern schedule
updateSchedule(PatternSchedule schedule)	Allows to change schedule settings
removeSchedule(Guid guid, boolean forceToDefault)	Deletes schedule with specified guid. When schedule is already associated with any patterns, the <code>forceToDefault</code> parameter allows you to control whether those patterns are switched to a default schedule or whether the method fails.
removeSchedule(Guid guid)	Shorthand method for removing pattern schedule, equals to <code>removeSchedule(guid, false)</code>
getDefaultSchedule()	Retrieves details of default schedule

Table 25. Pattern schedule management (continued)	
Method	Description
changeDefaultSchedule(String expression, boolean isCronExpression)	Allows changing default schedule expression. The <code>isCronExpression</code> flag defines whether expression parameter is a cron expression or interval expression (such as 1w 2d 3h 4m)
changeDefaultSchedule(int interval)	Allows changing default schedule by passing interval in minutes.
getExecutionGroupsInfo()	Retrieves information about all execution groups that are defined in schedules along with details of threads available on each storage server for each group

## Manual control of patterns execution

To grouping pattern methods provide ways of listing patterns that are being currently executed, and details of their next executions. The methods also provide means for the 'ad hoc' start of patterns processing, or interruption of the ones that are being executed.

Even though the API methods are called against a particular storage server, the actual processing is requested against respective execution groups across storage servers' thread pools. Therefore:

- Information that is related to processing status can be obtained from any server and always contains complete information (even if the particular storage server does not provide resources for the execution groups).
- Actual processing of the pattern might occur on a storage server other than the one used to request that processing.
- Pattern execution can be interrupted from any storage server even if a pattern is being processed by a server other than the one used to cause the interruption.

The following table lists pattern execution management methods and provides the descriptions of those methods:

Table 26. Pattern execution management methods	
Method	Description
getPatternRunInfo(Guid guid)	For a particular pattern, <code>guid</code> provides information that is related to the pattern execution (such as the next run time).
getPatternsRunInfo()	Provides information that is related to pattern execution for all patterns that are defined in TADDM database.
getPatternsInRun()	Provides information about all patterns that are being processed at the moment along with run information such as actual storage server on which pattern is being processed.

Table 26. Pattern execution management methods (continued)

Method	Description
runPatternNow(GroupingPattern pattern, boolean waitForCompletion); runPatternNow(String patternName, boolean waitForCompletion); runPatternNow(Guid patternGuid, boolean waitForCompletion)	Set of methods that allow you to start the pattern execution immediately. Pattern can be referred as (respectively): <ul style="list-style-type: none"> <li>• pattern object (obtained earlier via, for example, DataApi)</li> <li>• pattern name</li> <li>• pattern GUID</li> </ul> The <code>waitForCompletion</code> flag determines whether the method waits until the pattern is processed or finish just after requesting execution.
runPatternsNow(GroupingPattern[] patterns, boolean waitForCompletion); runPatternsNow Guid patternsGuids[], boolean waitForCompletion)	These methods are similar to <code>runPatternNow</code> , but allow the running of multiple patterns. The <code>waitForCompletion</code> flag determines whether a function waits until the last pattern processing is finished, or finish just after all pattern executions were requested.
runAllPatternsForGroup(String executionGroupId, boolean waitForCompletion)	Allows you to run all patterns that are associated with schedules from a particular execution group (typically this is a quick way of processing all of the patterns)
interruptPatternNow(String patternName); interruptPatternNow(Guid patternGuid); interruptPatternNow(GroupingPattern pattern)	Set of methods that allow you to stop processing of a particular pattern (if the pattern is being processed at the moment)
refreshJobs()	Refreshes and restarts all schedules for all patterns that are defined in TADDM database.

## Creating a grouping pattern

To create a grouping pattern, complete the following steps:

1. Create a Grouping Pattern object and set the values for the name and `hierarchyType` fields.
2. Create an array of selector objects and set their parameters to define the collection creation rules.
3. Attach all selectors to the grouping pattern by setting the selector `parent` field with the grouping pattern and setting grouping pattern selectors to the array of selectors.
4. Create pattern by using the grouping pattern API method **`createGroupingPattern`**.

## Deleting a grouping pattern

To delete grouping patterns, use a **`removeGroupingPattern`** command in grouping pattern API. The command accepts the `guid` parameter, which uniquely identifies the grouping pattern. Deleting grouping pattern also deletes all its attached selectors.

## Example: creating, retrieving, and deleting grouping patterns and selectors

The following example creates a grouping pattern and its selectors:

```
# ----- Begin sample code -----
import sys
import java

from java.io import *
from com.collation.platform.util import ModelFactory
```

```

from com.collation.proxy.api.client import *
from com.ibm.cdb.api import ApiFactory
from java.lang import System
from java.lang import String
from java.lang import Boolean
from java.lang import Class

from jarray import array

false = Boolean(0)
conn = ApiFactory.getInstance().getApiConnection("localhost", -1, None, false)
sess = ApiFactory.getInstance().getSession(conn, "administrator",
"collation", ApiSession.DEFAULT_VERSION);
api = sess.createCMDBApi()

# create Grouping pattern with name "Grouping Pattern 1"
name = "Grouping Pattern 1"
gpattern = ModelFactory.newInstance(Class.forName("com.collation.platform.
model.topology.customCollection.GroupingPattern"))
gpattern.setName("Grouping Pattern 1")
gpattern.setHierarchyType("BusinessApplication")

SelectorClass = Class.forName("com.collation.platform.model.topology.
customCollection.Selector")
selector1 = ModelFactory.newInstance(SelectorClass)
selector1.setType(0) # type MQL
selector1.setName("Linux Computer systems")
selector1.setQuery("LinuxUnitaryComputerSystem")
selector1.setUseTraversalTemplate(false)
selector1.setGroupingNameExpression("LinuxComputerSystems")
selector1.setParent(gpattern)

selector2 = ModelFactory.newInstance(SelectorClass)
selector2.setType(0) # type MQL
selector2.setName("Windows Computer systems")
selector2.setQuery("WindowsComputerSystem")
selector2.setUseTraversalTemplate(false)
selector2.setGroupingNameExpression("WindowsComputerSystems")
selector2.setParent(gpattern)

gpattern.setSelectors(array([selector1, selector2], SelectorClass))

guid=api.createGroupingPattern(gpattern)

# retrieve stored grouping pattern
gpattern = api.getGroupingPattern(guid)

print 'Grouping pattern: ', gpattern
for sel in gpattern.getSelectors():
    print '    Selector:', sel

# retrieve all grouping patterns
allPatterns = api.getAllGroupingPatterns()

for pat in allPatterns:
    print 'Grouping pattern: ', pat
    for sel in pat.getSelectors():
        print '    Selector:', sel

# remove grouping pattern
api.removeGroupingPattern(guid)

api.close()
sess.close()
System.exit(0)

# ----- End sample code -----

```

## Example API MQL queries

The following examples show you how to search for various patterns by using API MQL queries.

### Finding all business applications for a particular grouping pattern

```
api.sh -u user -p pass find "select * from CustomCollection where
groupingPattern.guid == '9C704FF849993840B89FBECA5E183AFA'"
api.sh -u user -p pass find "select * from CustomCollection where
groupingPattern.name == 'GP'"
```

### Finding all nodes and paths for a particular business application

```
api.sh -u user -p pass find "select * from Node where
parent.guid=='B71A946FEA753FB38B57B65775DA6519'"
api.sh -u user -p pass find "select * from Path where
parent.guid=='B71A946FEA753FB38B57B65775DA6519'"
```

### Finding all nodes that point to a particular CI

```
api.sh -u user -p pass find "select * from Node where
sourceGuid == '785614419CED31ACB24989A24F8ED52A'"
api.sh -u user -p pass find "select * from Node where
displayName contains 'NC9128109078'"
```

### Finding a business application compatible with earlier versions generated from a new business application

First, use the following query to gather some initial data basing on the new business application:

```
api.sh -u user -p pass find "select backwardCompatibleBusinessAppGuid,
hierarchyType, originalBusinessAppType from CustomCollection
where guid == 'B71A946FEA753FB38B57B65775DA6519'"
```

As a result, you get the GUID of the application compatible with earlier versions and information about which class to query. For a migrated business application, the `originalBusinessAppType` field contains this information, otherwise the `hierarchyType` points to it.

With such information, you can query the application compatible with earlier versions, by using the following query:

```
api.sh -u user -p pass find "select * from Application where
guid == '9EA1FAD9B4153000BD21CB2967ADB3DB'"
```

## Loading grouping patterns

The bulk load program can be used to load grouping patterns into TADDM database. When grouping patterns are loaded into the TADDM database, they can be inspected and optionally modified in Data Management Portal.

The bulk load program supports loading IdML books with GroupingPattern and Selector objects. When books contain GroupingPattern and Selector objects, the following limitations apply:

- The bulk load program must operate in the graph mode, that is, the `-g` command line option must be specified.
- Each book must fit bulk loader cache. This means that the whole IdML book that contains GroupingPattern and Selector objects must be loaded at one time. If your books are too large, split them or increase the bulk load cache. Cache size is defined by the `com.ibm.cdb.bulk.cachesize` property in the `$COLLATION_HOME/etc/bulkload.properties` file.

See also the *Bulk load program* topic in the *TADDM User's Guide*.

## Grouping name expression

The `GroupNameExpression` field of the selector is used to generate the grouping name expressions of the custom collections, for example business applications that are generated by the `BizAppsAgent`.

The appropriate custom collection is built for each core CI that is defined by the selector's query. The grouping name expression of the custom collection is generated for each core CI.

The advanced customization of the group can be allowed by defining the `GroupNameExpression` field as a pattern that can extract attribute values from the core CIs. This pattern can also perform a regular expression matches on the core CIs instead of just providing static text names. The `GroupNameExpression` supports the limited Apache Velocity syntax to provide such flexibility in defining grouping name expressions of the generated custom collections.

**Tip:** For details about model objects, for example their attributes and relationships, see the CDM documentation that is shipped with TADDM. It is in the `cdm/datadictionary/cdm/misc/CDM.htm` directory.

There are two predefined variables in the `GroupNameExpression` pattern:

### **\$scoreCI**

This variable represents the core configuration item that is being processed by the `BizAppsAgent`. It can be used to extract attributes of the core CI. For example, `$scoreCI.displayName` gets the `displayName` attribute from each core CI and puts its value into the generated grouping name expression.

### **\$utils**

This variable represents the utilities available in patterns. The following utility methods are available:

- `$utils.regex(inputText, regexPattern1 [, ..., regexPatternN])` - a method that extracts the part of the `inputText` attribute by using regular expression patterns that are defined in the `regexPattern1..N` attributes. Extraction is performed by using a regular expression matching groups. Therefore, `regexPattern1..N` attributes must define at least one matching group. Regular expression patterns without any matching groups are invalid and result in an error.

**Note:** Not matching groups, for example `"abc(?:\d+)"`, are ignored. Therefore, this example is invalid, as it does not contain at least one matching group.

The contents of the first matching group that successfully matched the input text is selected as a result. For example, for the input text `"abc-1234-def"` and pattern `"[A-Z]+-(\d+)|[a-z]+-\d+(\w+)"`, the second matching group `(\w+)` that captures the `"def"` substring is selected because the first one does not match the input.

The `regexPattern1..N` regular expression patterns are matched in sequence 1 to N until the first match is found.

- `$utils.or(expression1, ..., expressionN)` - a method that evaluates all expressions and selects the first result of the expression that is not null as its result.

For example, if the `$scoreCI.description` property has no value and the `$scoreCI.name` property is set to a valid value, then `utils.or($scoreCI.description, $scoreCI.name)` returns the result of the `$scoreCI.name` expression as its result. The error that the `$scoreCI.description` property is not available is not reported, as it would be, if only the `$scoreCI.description` was used.

- `$utils.toUpper(inputText)` - a method that converts the `inputText` attribute value to uppercase characters.
- `$utils.toLower(inputText)` - a method that converts the `inputText` attribute value to lowercase characters.
- `$utils.trim(inputText)` - a method that removes all leading and trailing white space characters from the `inputText` attribute value.
- `$utils.replace(inputText, pattern, substitute)` - a method that replaces all substrings of the `inputText` attribute that match the pattern with substitute. The pattern attribute is a regular expression pattern, which can include matching groups. The substitute

attribute can reference the matching groups that are defined in the pattern attribute by using \$1 for the first matching group, \$2 for the second matching group, and so on.

**Note:** If the dollar sign (\$) or backslash (\) are used in the replacement string, they must be escaped by adding another backslash as a prefix to the string. For example, in the `$utils.replace($coreCI.name, "some pattern", "abc\\$123")` invocation, the substitute is a static text "abc\$123" that includes the dollar sign because the dollar sign was escaped by using the `\\$` combination (therefore \$123 does not reference the matching group 123). The substitute texts that contain the dollar sign or backslash alone, for example "abc\$def" or "abc\\def", are invalid because the dollar sign and backslash were not escaped by adding the backslash as a prefix. In this case, the valid substitute texts are "abc\\\$def" and "abc\\\\def".

## Examples

- For a grouping pattern with the following elements:

- hierarchyType: BusinessApplication,
- selector with MQL query: `SELECT * FROM Database`,
- GroupingNameExpression: `$coreCI.displayName`,

the generated grouping name expression is the value of the `displayName` attribute of each core CI. If display names are unique, separate business applications with grouping name expression set to the display name of that core CI are created for every core CI selected by the selectors' query.

- For a grouping pattern with the following elements:

- hierarchyType: BusinessApplication,
- selector with MQL query: `SELECT * FROM J2EEApplication WHERE name contains '_EDT- '`,
- GroupingNameExpression: `eDayTrader`,

a single business application with the `eDayTrader` grouping name expression that contains all core CIs and their dependent objects is created.

- For a grouping pattern with the following elements:

- hierarchyType: BusinessApplication,
- selector with MQL query: `SELECT * FROM Database WHERE name starts-with 'prod_'`,
- GroupingNameExpression: `BizApp- $\{utils.regex("\\{coreCI.name}\\", "prod_(.*)")\\}- $\{coreCI.label\\}$$` ,

and for the core CI with the "prod\_EDT1" name and the "trader1" label, the generated grouping name expression is `BizApp-EDT1-trader1`.

All partial custom collections that are generated for core CIs with a matching name, for example EDT1, and a matching label, for example trader1, are assigned into a single business application collection with grouping name expression, for example `BizApp-EDT1-trader1`. It allows for dynamic assignment and grouping of configuration items and their related objects to business applications based on any matching attribute combinations.

## Apache Velocity syntax

The `GroupingNameExpression` pattern supports limited Apache Velocity syntax and is validated during creation. The following validations are enforced:

- No `#include` or `#parse` directives are allowed in the pattern.
- `$coreCI` and `$utils` variables are predefined and ready to be used in the pattern, but they cannot be redefined. Other variables can be defined and used in the pattern by using the `#set` directive.
- On the `$coreCI` variable, only property access is allowed. No method calls, for example, getters and setters, are allowed. The result of property access to the `$coreCI` variable must be a string or other



simple data type, like Boolean, int, long, float, double. Especially, it cannot be a model object or array. Accessing the `$scoreCI` properties that are model objects or arrays to get their properties or elements is allowed, when the result is not a model object or array.

Examples:

- `$scoreCI.name` is allowed.
- `$scoreCI.OSRunning.name` is allowed. It is used to access `OSRunning`, which is a model object of the `OperatingSystem` type, to get the name property.
- `$scoreCI.OSInstalled[1].name` is allowed. It is used to access the second element of an array of `OperatingSystem` elements to get the element name property.
- `$scoreCI` is not allowed. `$scoreCI` is a model object.
- `$scoreCI.OSRunning` is not allowed.
- `$scoreCI.OSInstalled` is not allowed. It results in an array of model objects.
- `$scoreCI.OSInstalled[1]` is not allowed. The result of `OSInstalled[1]` is a model object `OperatingSystem`.
- `$scoreCI.setName("test")` is not allowed,
- `$scoreCI.getName()` - is not allowed.
- `$scoreCI.hasName()` - is not allowed. However, the equivalent construct `#if ($scoreCI.name) ... #end` is allowed. Additionally, `$utils.or($scoreCI.name, $scoreCI.otherProperty, ...)` can be used to select the first property that exists and that is set.
- `$scoreCI.test()` - is not allowed. Any other method call is not allowed as well.
- On the `$utils` variable only defined method calls are allowed:
  - `$utils.regex(inputText, regexPattern1 [, ..., regexPatternN])`

Null values are not allowed on the `regexPattern1..N` attributes. Each `regexPattern` must define at least one matching group. The `inputText` attribute can be null, when the `$utils.regex` invocation is not the last invocation in chain resulting in insertion of its result into the final text. For example, if such `$utils.regex` call trying to access not set `$scoreCI` property is nested inside the `$utils.or` invocation, it does not generate error when `$utils.or` is able to select any other expression value that is not null as a result. In all other cases, such `$utils.regex` call results in an error.
  - `$utils.toUpper(inputText)`
  - `$utils.toLower(inputText)`
  - `$utils.trim(inputText)`
  - `$utils.replace(inputText, pattern, substitute)`

The `pattern` and `substitute` attributes must be valid strings that are not null. The `pattern` attribute must be a valid regular expression pattern. The `substitute` attribute must be a valid substitution or replacement pattern with appropriate escaping of the dollar sign (\$) and backslash (\), and the appropriate usage of matching groups.
  - `$utils.or(expression1, ..., expressionN)`

Expression attributes can be null, when the whole expression that involves the `$utils.or` invocation evaluates to a value that is not null.

The following list contains exemplary expressions, where the `$scoreCI.name` property is valid, the `$scoreCI.description` property is not set, and the `$scoreCI.instanceID` property does not exist. The `$scoreCI.instanceID` property can be a property that does not exist in case when the current `$scoreCI` is of the `ComputerSystem` type, which does not have the `instanceID` property (the `instanceID` property is a valid property of `AppServers`, and so on).
  - `$utils.or($scoreCI.description, $scoreCI.instanceID)` - not valid. It results in an error because it evaluates to a null value.

- `$utils.or($coreCI.description, $coreCI.instanceID, "default")` - valid. Even though the `description` property is not set and the `instanceID` property does not exist, the last attribute is a static text "default" and the whole `$utils.or` invocation successfully evaluates to this value.
- `$utils.or($coreCI.description, $coreCI.instanceID, $utils.name)` - valid. Even though the `description` property is not set and the `instanceID` property does not exist, the `name` property is valid and the whole `$utils.or` invocation evaluates to a valid value of `$coreCI.name`.
- `$utils.or($utils.or($coreCI.description, $coreCI.instanceID), "default")` - valid. Even though the inner invocation `$utils.or($coreCI.description, $coreCI.instanceID)` evaluates to an invalid value, the outer `$utils.or` invocation, and at the same time the whole expression, evaluate to a valid value "default".
- `$utils.or($utils.toUpper($coreCI.description), $utils.regex($coreCI.instanceID, "prod_(.*)", $utils.toLower($coreCI.name)))` - valid. Even though the `$utils.toUpper($coreCI.description)` and `$utils.regex($coreCI.instanceID, ...)` invocations are not valid, the outer `$utils.or` invocation, and at the same time the whole expression, evaluates to a valid value of `$utils.toLower($coreCI.name)`.

### Examples

- `BizApp-$utils.or($utils.regex($utils.toLower($coreCI.label), "appserver-(.+)"), $regex($coreCI.description, "application server number (\d+)"), "default")`
  - For the { `label="AppServer123"`, `name="CS-123"`, `description="..."` } core CI, the result is "BizApp-123" - extracted from `$coreCI.label` attribute.
  - For the { `label=null`, `name="CS-123"`, `description="This is an application server number 123. Installed mod...."` } core CI, the result is "BizApp-123" - extracted from `$coreCI.description` attribute.

The following `$utils.or` method calls can be nested:

- `BizApp-$utils.or($utils.or($coreCI.name, $coreCI.label, $utils.regex($coreCI.description, "name: (\S+)")), "default")`
- `BizApp-$utils.or($utils.regex($coreCI.name, "prod_(.*)|test_(.*)"), $utils.regex($coreCI.keyName, "systemA-(\\d+)|systemC-(\\d+)|system?-?- (.*)"), "default")`
- `BizApp-$utils.or($utils.regex($coreCI.name, "test_(.*)", "test1_(.*)"), $utils.regex($coreCI.keyName, "prod1_(.*)", "prod_(.*)"), "default")`
- `BizApp-$utils.or($coreCI.primarySAP.fqdn, $coreCI.primarySAP.primaryIpAddress.dotNotation, $coreCI.primarySAP.primaryIpAddress.byteNotation, $coreCI.primarySAP.primaryIpAddress.stringNotation, "unknown")`
- Equivalent to the preceding one `#set($ip = $coreCI.primarySAP.primaryIpAddress) BizApp-$utils.or($coreCI.primarySAP.fqdn, $ip.dotNotation, $ip.byteNotation, $ip.stringNotation, "unknown")`

## Displaying business applications

Once you generate your business applications, you can view them in the **Discovered Components** window.

1. Open Data Management Portal.
2. In the **Discovered Components** window, click **Groups** and then **Business Applications**.
3. Select the application that you want to display.

4. From the Actions list, select **Show Topology**.

## Configuring the maximum size of displayed topologies

Displaying large topologies is memory consuming and might even lead to a crash of the TADDM server. To prevent that, a safety check is implemented. You can also configure the maximum size of the topology, depending on your needs, in the `collation.properties` file.

The size of topology is defined by the number of nodes. It is determined dynamically depending on the current maximum Java heap size settings for Tomcat JVM process (TADDM 7.3.0) or for Liberty JVM process (TADDM 7.3.0.1, and later). The settings are defined based on a linear function  $(25 * M / 32) - 200$ , where M is the maximum Java heap size. For example:

- 600 topology nodes are allowed on 1 GB heap, which is a default setting.
- 3000 nodes are allowed on 4 GB heap.

**Note:** This limit is related to all topologies that are simultaneously displayed in multiple browsers that are connected to Data Management Portal on a single TADDM server. The limit does not apply only to the current user. Memory is freed when you log out, open another topology, or when a session expires.

When the topology that you want to display is too large, the following warning message is displayed in the place of the topology:

The requested graph has exceeded the number of allowed nodes.

To prevent such errors, increase the maximum heap size for Tomcat JVM process (TADDM 7.3.0) or for Liberty JVM process (TADDM 7.3.0.1, and later) by changing the value of the `com.collation.Tomcat.jvmargs` property (TADDM 7.3.0) or the `com.collation.Liberty.jvmargs` property (TADDM 7.3.0.1, and later).

## Configuration options

Use the following properties to customize the maximum Java heap size.

### **com.collation.topology.maxnodes**

Defines the maximum number of nodes that can be viewed in a topology. This property defines the topology size more precisely than the default settings. Set this property in the `collation.properties` file, when the default settings are not sufficient to display topologies.

Set the value in the following format: 1000. In this example, the maximum number of nodes is 1000. Values that are too high might lead to Out of Memory errors. If some topologies cause Out of Memory errors, set this property to a lower value.

### **com.collation.Tomcat.jvmargs (TADDM 7.3.0 only)**

Defines JVM options for Data Management Portal. This property can be used to define the maximum heap size. Set this property in the `collation.properties` file, when the default settings are not sufficient to display topologies.

Set the value in the following format: `-Xmx2048M`. In this example, the maximum heap size is 2048 MB (2 GB). You can use any value.

After you change the property, restart the TADDM server.

### **Fix Pack 1 com.collation.Liberty.jvmargs**

Defines JVM options for Data Management Portal. This property can be used to define the maximum heap size. Set this property in the `collation.properties` file, when the default settings are not sufficient to display topologies.

Set the value in the following format: `-Xmx2048M`. In this example, the maximum heap size is 2048 MB (2 GB). You can use any value.

After you change the property, restart the TADDM server.

## Processing of grouping patterns

You can control the mechanism of grouping patterns processing, and therefore control the process of generating business applications.

### Grouping patterns configuration

You can control the process of business application creation by using grouping patterns configuration. The configuration allows you to include or exclude chosen relations during data traversal and chosen classes from resulting business applications. You can also change the dependency direction for relations that are defined in Common Data Model, and assign tiers to business application elements.

Grouping patterns configuration is stored in an XML format in the database. You can export the configuration to the XML file, and import it from the XML file.

Grouping patterns can be assigned to the default configuration and a customized configuration. The default configuration applies to all grouping patterns and is loaded when TADDM is started, but the customized configuration has a higher priority. It means that when a grouping pattern is attached to an customized configuration, the default configuration applies in all cases that are not specified by the customized grouping pattern configuration. For more details about creating customized configurations, see [“Attaching a custom configuration to a grouping pattern” on page 210](#).

The XML configuration file consists of the following sections:

#### **general**

contains configuration that defines additional logging level details and maximum hops number.

#### **compositionConfiguration**

defines these elements that are visible as business application elements. The following subsections are available:

- includeInComposition
- excludeFromComposition

#### **traversalConfiguration**

allows you to exclude or include particular relations or dependencies during creation process of business applications. The following subsections are available:

- excludedRelationships
- includedRelationships

#### **tiers**

is used to create functional group names in business applications that are compatible with earlier versions.

#### **directions**

denotes dependency direction for relations that are defined in Common Data Model. It can be a part of the default configuration only.

The following configuration is an excerpt from the default configuration:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xml>
  <tiers>
    ...
    <tier>
      <name>Computer Systems</name>
      <rule>
        <className>ComputerSystem</className>
      </rule>
    </tier>
    ...
  </tiers>
  <traversalConfiguration>
    <excludedRelationships>
      <exclude relation="{any}" source="customCollection.CustomCollection"
target="{any}"/>
      <exclude relation="{any}" source="customCollection.GroupingPattern"
```

```

target="{any}"/>
    <exclude relation="{any}" source="{any}"
target="customCollection.GroupingPattern"/>
    <exclude relation="{any}" source="customCollection.Path"
target="{any}"/>
...
    </excludedRelationships>
</traversalConfiguration>
<compositionConfiguration>
    <includeInComposition>
        <include type="simple.SComputerSystem"/>
        <include type="simple.SDeployableComponent"/>
        <include type="simple.SFunction"/>
        <include type="simple.SGroup"/>
        <include type="simple.SSoftwareServer"/>
    </includeInComposition>
    <excludeFromComposition>
        <exclude type="customCollection.GroupingPattern" />
        <exclude type="app.FunctionalGroup" />
    </excludeFromComposition>
</compositionConfiguration>
<directions>
    <forwardRelationships>
        <forward relation="only relation.Provides"
source="sys.blade.BladeCenterManagementModule" target="sys.blade.Alert"/>
        <forward relation="only relation.Provides"
source="sys.vmware.VMWareVirtualSwitch" target="sys.vmware.VMWarePortGroup"/>
        <forward relation="only relation.Provides"
source="app.AppServer" target="app.JVM"/>
    ...
    </forwardRelationships>
    <reverseRelationships>
        <reverse relation="only app.dependencies.SwitchToDevice"
source="{any}" target="{any}"/>
        <reverse relation="only relation.ControlsAccess"
source="{any}" target="{any}"/>
        <reverse relation="only relation.Contains"
source="{any}" target="{any}"/>
    ...
    </reverseRelationships>
</directions>
</xml>

```

For each type of configuration, a source, target, or relation class can be defined in the following ways:

- Any type ("{any}") - it matches any class from Common Data Model.
- Class name - it matches a particular class and all of its subclasses.
- Class name only (only) - it matches a particular class only, without any of its subclasses.

Examples:

- `<exclude relation="{any}" source="sys.ComputerSystem" target="{any}"/>`

Excludes from traversal all types of relations from ComputerSystem and all its subclasses to any target class from Common Data Model.

- `<exclude relation="only relation.InvokedThrough"
source="app.messaging.mq.MQLocalQueue"
target="app.messaging.mq.MQChannel"/>`

Excludes from traversal a relation InvokedThrough (but not its subclasses) between MQLocalQueue and MQChannel (and their subclasses).

You can provide a class name as a short class name or a fully qualified class name. For example, you can use `sys.linux.LinuxUnitaryComputerSystem` instead of `com.collation.platform.model.topology.sys.linux.LinuxUnitaryComputerSystem`. Class name is not case-sensitive.

## General configuration

The general section of the configuration file defines additional logging level details and maximum hops number.

### Example:

```
<general>
  <maxHopsLimit>10</maxHopsLimit>
  <firstTierOnly>true</firstTierOnly>
  <infoLevel>GENERAL</infoLevel>
</general>
```

The general configuration consists of the following parameters:

#### maxHopsLimit

This parameter defines how far from the core CI the composition engine can go during building an application. The value defines the number of elements.

**Important:** The elements that are not visible as the elements of the business application are not skipped. For more information, see [“Composition configuration” on page 205](#).

The default value of this parameter is 10.

#### Fix Pack 1 firstTierOnly

This parameter specifies how many tiers are processed in search for specified elements. This parameter applies only to the XML configuration of the grouping pattern. The elements that are found by the selector of the grouping pattern are assigned to tiers based on the conditions specified by the tiers. When this parameter is set to `true`, and the elements match the conditions of a tier, the remaining tiers are ignored. When this parameter is set to `false`, even when elements that match conditions of one tier are found, other tiers are also processed. Therefore, it is possible that one element is assigned to many tiers, if it matches all tiers conditions.

The default value of this parameter is `true`.

#### infolevel

This parameter specifies the details level of logs when the INFO level is set. For example, it allows you to obtain more details for business applications in the log files without switching to the DEBUG level. The following levels are available:

- NOINFO - no information.
- DEFAULT - information about starting and stopping a business application only.
- GENERAL - the DEFAULT information and core CI data.
- DETAILS - the GENERAL information and composition engine traversal route details.
- MAXINFO - maximum detail level, currently the same as for the DETAILS level.

## Traversal configuration

Traversal configuration consists of included and excluded relations between objects.

By default, all existing CIs and their relations are traversed during business application composition. Business applications are composed during the grouping pattern processing. However, if you use traversal configuration, some relations can be skipped from processing. It means that the compositor does not go further through excluded relationships. It applies to either implicit relations, which are relations that are defined in Common Data Model, or explicit relations, which can link any CIs and can be created by using TADDM API or UI. However, explicit relations that are generated for implicit relations by using the `explicitrel.sh` script are ignored by default, and it cannot be configured.

In the following example, first two `relation` tags exclude completely the whole CI type by excluding all incoming and outgoing relations. The third `relation` tag excludes only one simple relation.

### Example:

```
<traversalConfiguration>
  <excludedRelationships>
    <exclude relation="{any}" source="admin.AdminInfo">
```

```

target="{any}" />
    <exclude relation="{any}" source="{any}"
target="admin.AdminInfo" />
    <exclude relation="only relation.DeployedTo" source="app.j2ee.
J2EEApplication" target="app.j2ee.websphere.WebSphereCell" />
    ...
    <includedRelationships>
    <include relation="{any}" source="admin.AdminInfo"
target="{any}" />
    <include relation="{any}" source="{any}"
target="admin.AdminInfo" />
    ...

```

The traversal configuration consists of the following parameters:

### **excludedRelationships**

This parameter contains a series of excluded relationships.

### **includedRelationships**

This parameter contains a series of included relationships. Although all relationships are included by default, this parameter is useful in case of grouping pattern configuration, when you want to include a relation that is excluded in the default configuration.

The `excludedRelationships` and `includedRelationships` parameters contain the following elements:

#### **relation**

The name of the relation or the dependency type.

#### **source**

The name of the relation source object type. If you do not want to define any specific source, you can set the value to `{any}`.

**Fix Pack 2** In TADDM 7.3.0.2, and later, you can also add the `hierarchyType` attribute of a model object to the source of the relation to be more specific. After the name of the relation source object type, add a colon (:) and the value of the `hierarchyType` attribute. For example, `source="app.AppServer:IBMTivoliEnterpriseConsole"`.

To make sure about the correct value of the `hierarchyType` attribute, you can query CI of a given type by using the TADDM APIs.

**Fix Pack 3** In TADDM 7.3.0.3, and later, you can also add the `hierarchyDomain` attribute of a model object to the source of the relation to be more specific. After the name of the relation source object type, add a colon (:) and the value of the `hierarchyDomain` attribute. When you are applying this filter, you must also add the value of the `hierarchyType` attribute. For example, `source="simple.SSoftwareServer:app.placeholder.client.remote.Unknown"`, where `app.placeholder.client.remote` is the value of the `hierarchyDomain` attribute, and `Unknown` is the value of the `hierarchyType` attribute. The `hierarchyType` attribute is always specified at the end, and is separated from the `hierarchyDomain` attribute with a dot. You can also use an asterisk (\*), which stands for one or more full parts of the domain name, or for the `hierarchyType` attribute. Examples:

```

source="simple.SSoftwareServer:app.placeholder.*.Unknown"
source="simple.SSoftwareServer:app.placeholder.client.remote.*"

```

#### **target**

The name of the relation target object type. If you do not want to define any specific target, you can set the value to `{any}`.

**Fix Pack 2** In TADDM 7.3.0.2, and later, you can also add the `hierarchyType` attribute of a model object to the target of the relation to be more specific. After the name of the relation target object type, add a colon (:) and the value of the `hierarchyType` attribute. For example, `target="app.AppServer:MySQL"`.

To make sure about the correct value of the `hierarchyType` attribute, you can query CI of a given type by using the TADDM APIs.

**Fix Pack 3** In TADDM 7.3.0.3, and later, you can also add the `hierarchyDomain` attribute of a model object to the target of the relation to be more specific. After the name of the relation target object type, add a colon (:) and the value of the `hierarchyDomain` attribute. When you are applying this filter, you must also add the value of the `hierarchyType` attribute. For example, `target="simple.SSoftwareServer:app.placeholder.server.local.Java"`, where `app.placeholder.server.local` is the value of the `hierarchyDomain` attribute, and `Java` is the value of the `hierarchyType` attribute. The `hierarchyType` attribute is always specified at the end, and is separated from the `hierarchyDomain` attribute with a dot. You can also use an asterisk (\*), which stands for one or more full parts of the domain name, or for the `hierarchyType` attribute. Examples:

```
target="simple.SSoftwareServer:*.placeholder.*.Java"
target="simple.SSoftwareServer:app.placeholder.*"
```

#### **Fix Pack 2** **direction**

The direction of the dependency traversal defined for a specific relationship. The following values are available:

- **UP:** The exclusion or inclusion rule is applied only when the current traversal direction is up the dependency chain, starting from the source object type.
- **DOWN:** The exclusion or inclusion rule is applied only when the current traversal direction is down the dependency chain, starting from the source object type.
- **UP\_AFTER\_DOWN:** The exclusion or inclusion rule is applied only when the current traversal direction is down and then up the dependency chain, starting from the source object type. It is the equivalent of the `LowerUp` option that you can select in Data Management Portal.
- **DOWN\_AFTER\_UP:** The exclusion or inclusion rule is applied only when the current traversal direction is up and then down the dependency chain, starting from the source object type. It is the equivalent of the `HigherDown` option that you can select in Data Management Portal.

Traversal configuration is related to relation direction as defined in Common Data Model (relations direction configuration is not taken under consideration). However, in case of explicit dependencies traversal does take direction into consideration because the source and target class is not constrained by Common Data Model (any class can be used as a source or target of explicit relation).

## More examples

- Skipping a specific relation between a pair of specific objects.

```
<exclude relation="relation.RunsOn" source="sys.OperatingSystem"
target="sys.ComputerSystem"/>
```

- Skipping any relations, for which a particular class is a source.

```
<exclude relation="{any}" source="net.BindAddress" target="{any}"/>
```

- Skipping any relations, for which a particular class is a target.

```
<exclude relation="{any}" source="{any}" target="net.BindAddress"/>
```

- Skipping a relation but not its subclasses.

**Note:** `dev.RealizesExtent` is a subclass of `relation.Realizes` and it is processed even though `relation.Realizes` is skipped.

```
<exclude relation="only relation.Realizes" source="sys.FileSystem"
target="sys.FileSystem"/>
```

- **Fix Pack 2** Skipping any relation, for which IBM Tivoli Enterprise Console object of the AppServer object type is a source.

```
<exclude relation="{any}" source="app.AppServer:IBMTivoliEnterpriseConsole"
```



```
target="{any}" />
```

**Note:** For more information about hierarchyType attribute of custom server templates, see [“Creating and managing custom server templates” on page 16.](#)

- **Fix Pack 2** Skipping a specific relation between a pair of specific objects. Additionally, the relation is skipped only when it is found while traversing the objects down the dependency chain.

```
<exclude relation="relation.RunsOn" source="app.AppServer"
target="sys.ComputerSystem" direction="DOWN"/>
```

If this exclusion rule is added to the patten configuration, which is attached to a pattern where ComputerSystem CI is a core CI (traversal starting point), then all AppServers (applications) that run on this core CI are added to the topology, because AppServers depend on ComputerSystem. Other AppServers that are connected with the already added AppServers can also be added to the topology through IpConnection dependencies. However, the exclusion rule provided in the example is applied when the traversal engine tries to add the host ComputerSystems on which these AppServers are running. In this case, the AppServer->RunsOn->ComputerSystem relation is traversed down and the exclusion rule is applied.

- **Fix Pack 3** Skipping any relation, for which the target is an object with SSoftwareServer type, and has the hierarchyDomain attribute set to app.placeholder.client.remote, and the hierarchyType attribute set to Unknown.

```
<exclude relation="{any}" source="{any}" target="simple.SSoftwareServer:
app.placeholder.client.remote.Unknown"/>
```

You can also configure BizAppsAgent to traverse only specific set of relations. For example:

```
<excludedRelationships>
<exclude source="{any}" target="{any}" relation="{any}" />
</excludedRelationships>
<includedRelationships>
<include target="{any}" source="sys.SystemPCComputerSystem"
relation="relation.Virtualizes" />
<include target="{any}" source="sys.linux.LinuxUnitaryComputerSystem"
relation="relation.Virtualizes" />
<include target="{any}" source="only sys.ComputerSystem"
relation="relation.Virtualizes" />
</includedRelationships>
```

## Composition configuration

Composition configuration consists of objects that are included in business applications and excluded from them.

If not specified otherwise in composition configuration, all objects are excluded from business application composition by default. The default configuration includes subclasses of high-level interfaces that include computer systems, software servers, and components that are deployed to them, functions and various kinds of groupings, for example, clusters, or cells. Business applications are affected by the composition configuration after they are built. The excluded objects are filtered out of the business application, but you can still view them in the Path details pane.

### Example:

```
<compositionConfiguration>
  <includeInComposition>
    <include type="simple.SComputerSystem" />
    <include type="simple.SDeployableComponent" />
    <include type="simple.SFunction" />
    <include type="simple.SGroup" />
    <include type="simple.SSoftwareServer" />
  </includeInComposition>
  <excludeFromComposition>
    <exclude type="customCollection.GroupingPattern" />
    <exclude type="process.AccessCollection" />
  </excludeFromComposition>
</compositionConfiguration>
```

The composition configuration consists of the following parameters:

### **includeInComposition**

This parameter contains a series of the `include` tags that specify the names of Common Data Model types. Only objects of the types that are included and their subtypes can be business application components.

### **excludeFromComposition**

This parameter contains a series of the `exclude` tags. It allows you to exclude some subtypes of the included types. For example, you can include the `SGroup` type branch with two exceptions, `GroupingPattern` and `AccesCollection`, like in the preceding example.

You can include all traversed CIs in the business application by specifying the `"{any}"` value for the `type` parameter in the `include` tag. See the following example:

```
<compositionConfiguration>
  <includeInComposition>
    <include type="{any}" />
  </includeInComposition>
</compositionConfiguration>
```

## ***Relations direction configuration***

Relations direction configuration denotes dependency direction for relations that are defined in Common Data Model.

**Note:** Relations direction configuration can be a part of the default configuration only. It cannot be changed for a specific grouping pattern.

Relations between CIs are always traversed based on the dependency directions. For example, `Application Server` always depends on the hosting `ComputerSystem` because it cannot run without it. If `ComputerSystem` is shut down, `Application Server` cannot function without it and is shut down too. But `ComputerSystem` does not depend on `Application Server` because its functioning is not affected by turning off `Application Server`. Relations in Common Data Model are not always aligned with objects dependency direction. The purpose of this type of configuration is to enable reversing particular relations during data traversal.

### **Example:**

```
<directions>
  <reverseRelationships>
    <reverse relation="only relation.GroupMemberOf" source="{any}"
target="{any}" />
    <reverse relation="only relation.Provides" source="{any}"
target="{any}" />
    ...
  <forwardRelationships>
    <forward relation="only relation.Provides" source="sys.blade.
BladeCenterManagementModule" target="sys.blade.Alert" />
    <forward relation="only relation.Provides" source="sys.vmware.
VMWareVirtualSwitch" target="sys.vmware.VMWarePortGroup" />
    ...
</forwardRelationships>
```

The relations direction configuration consists of the following parameters:

### **reverseRelationships**

This parameter is used to reverse Common Data Model relation directions. It means that the business application composition engine traverses through this relation in opposite direction than the direction defined in Common Data Model. You must provide values for the following parameters:

- **relation** - the name of Common Data Model relation or dependency.
- **source** - the name of the relation's source object type. In case of dependency, only `"{any}"` value is allowed.
- **target** - the name of the relation's target object type. In case of dependency, only `"{any}"` value is allowed.

## forwardRelationships

This parameter allows you to add some exception to the reversed relationships. In the preceding example, the Provides relation was reversed but with two exceptions that are defined in the forwardRelationships section.

If a particular relation type is not included in the default configuration, it is treated as aligned with objects' dependency. For example, since relation.Uses is by default aligned with its dependency, this element is not needed in the configuration:

```
<forward relation="relation.Uses" source="{any}" target="{any}"/>
```

However, the configuration must contain exceptions for a specific pair of classes, for example:

```
<reverse relation="only relation.Uses" source="app.db.oracle.OracleDataFile"
target="app.db.oracle.OracleTableSpace"/>
```

Direction configuration can also contain relation types that must be reversed for all its occurrences in Common Data Model. For example, relation Manages must be reversed globally:

```
<reverse relation="only relation.Manages" source="{any}" target="{any}"/>
```

However, any exceptions must be included in the forwardRelationships element:

```
<forward relation="only relation.Manages" source="sys.HMC"
target="sys.ComputerSystem"/>
```

You can change directions only for these classes for which a relation is defined. You cannot change a relation's direction for these subclasses of classes for which the relation is defined. For example, a relation sys.OperatingSystem -> relation.RunsOn -> sys.ComputerSystem cannot be reversed for sys.linux.Linux -> relation.RunsOn -> sys.linux.LinuxUnitaryComputerSystem. However, you can include or exclude relation traversal in that way.

## Tiers configuration

One of the purposes of creating customized grouping pattern configuration is to define the tiers which are assigned to the elements of created groups. The elements of each group that is created by BizAppsAgent can then be divided into tiers.

The configuration for a tier contains a tier name and a set of rules with conditions. The tiers definitions are processed in the defined order. The first tier for which at least one rule meets all conditions that are defined for a particular configuration item is assigned to an element.

**Note:** Fix Pack 1 You can control how many tiers are processed by using the firstTierOnly parameter, which is specified in the general section of the pattern configuration. For details, see [“General configuration”](#) on page 202.

The customized configuration that is specific to a particular grouping pattern is processed before the default configuration.

The following example shows the simple tier configuration with the tiers definition for elements in created groups. This is also the default tier configuration.

```
<tiers>
  <tier>
    <name>Computer Systems</name>
    <rule>
      <className>ComputerSystem</className>
    </rule>
  </tier>
  <tier>
    <name>App Servers</name>
    <rule>
      <className>AppServer</className>
    </rule>
  </tier>
</tiers>
```

The following example shows the advanced tier configuration with the tiers definition for elements in created groups.

```
<tiers>
  <tier>
    <name>My tier name</name>
    <rule>
      <className>ComputerSystem</className>
      <displayName type="wildcard">*.pl</displayName>
    </rule>
    <rule>
      <className type="strict">LinuxUnitaryComputerSystem</className>
      <displayName type="wildcard">*ibm*</displayName>
    </rule>
  </tier>
  <tier>
    <name>Windows Computer Systems</name>
    <rule>
      <className>WindowsComputerSystem</className>
    </rule>
    <rule>
      <className type="regexp">.*ComputerSystem</className>
      <expression>
        <pattern>$CI.OSRunning.OSName</pattern>
        <match type="regexp">.*Windows.*</match>
      </expression>
    </rule>
    <rule>
      <hierarchyType type="strict">WindowsComputerSystem</hierarchyType>
    </rule>
  </tier>
</tiers>
```

For more information about configuration elements, see [“Tiers configuration elements and attributes” on page 208](#).

### Related concepts

[“Business application tiers” on page 235](#)

Business application tiers are groups of similar business application elements. They are used to create functional groups to integrate TADDM with other products and to ensure compatibility with business applications that were created in TADDM 7.2.2.

#### *Tiers configuration elements and attributes*

Refer to the following table to learn about the elements and attributes of the tier configuration that you can customize.

Table 27. Tiers configuration elements and attributes.	
Element	Description and attributes
tiers	The root element for the tiers definition. Every tier definition must be placed under this element.
tier	The element for the tier definition.
name	(Required) The element for the tier name definition. It must be placed under the tier definition.
rule	The element for the tier rule definition. It must be placed under the tier definition. At least one rule is required. Multiple rules for one tier definition are allowed. In this case, the OR logical operator is included between them. The rule element requires at least one condition element: className, displayName, or hierarchyType. If the conditions are mixed, the AND logical operator is included between them.

Table 27. Tiers configuration elements and attributes. (continued)

Element	Description and attributes	
className	(Optional) The element for the tier rule condition against a class name. Only one element per rule is allowed. If more elements are required, you must create a new rule for the tier definition. Use only short class names, for example ComputerSystem.	
	type	<p>(Optional) Defines how the defined condition is treated. The following values are possible:</p> <ul style="list-style-type: none"> <li>• <b>strict</b> - the default value. Configuration Item (CI) CDM class must be the provided class or inherit from it. Only short class names are allowed, for example ComputerSystem.</li> <li>• <b>wildcard</b> - the short class name must match. This value allows using the asterisk sign (*) (zero or more characters) and the question mark (?) (zero or one character). Inheritance is ignored.</li> <li>• <b>regexp</b> - short class name must match the provided regular expression. Inheritance is ignored.</li> </ul>
displayName	(Optional) The element for the tier rule condition against a display name. Only one element per rule is allowed.	
	type	<p>(Optional) Defines how the defined condition is treated. The following values are possible:</p> <ul style="list-style-type: none"> <li>• <b>strict</b> - the default value. The display name must be equal to the provided value.</li> <li>• <b>wildcard</b> - the display name must match. This value allows using the asterisk sign (*) (zero or more characters) and the question mark (?) (zero or one character).</li> <li>• <b>regexp</b> - the display name must match the provided regular expression.</li> </ul>
hierarchyType	(Optional) The element for the tier rule condition against a hierarchy type. Only one element per rule is allowed.	
	type	<p>(Optional) Defines how the defined condition is treated. The following values are possible:</p> <ul style="list-style-type: none"> <li>• <b>strict</b> - the default value. The hierarchy type must be equal to the provided value.</li> <li>• <b>wildcard</b> - the hierarchy type must match. This value allows using the asterisk sign (*) (zero or more characters) and the question mark (?) (zero or one character).</li> <li>• <b>regexp</b> - the hierarchy type must match the provided regular expression.</li> </ul>
expression	(Optional) The element for the tier rule condition against a <b>CI</b> (Configuration Item) field. It requires the <b>pattern</b> and <b>match</b> elements to be defined. Only one element per rule is allowed. This condition is evaluated only when other conditions for the rule are true.	

Table 27. Tiers configuration elements and attributes. (continued)

Element	Description and attributes	
pattern	The element required by the expression to define the grouping name expression pattern to get the value of the <b>CI</b> (configuration item) field. The value of the <b>CI</b> field is extracted by using the limited Apache Velocity syntax, similar to the one used for groupId in grouping patterns. The only difference is that instead of the \$scoreCI variable name, the corresponding \$CI is required, for example \$CI.OSRunning.OSName.	
match	The element required by the expression to define a condition against a value that is extracted by using the specified pattern.	
	type	<p>(Optional) Defines how the defined condition is treated. The following values are possible:</p> <ul style="list-style-type: none"> <li>• <b>strict</b> - the default value. The extracted value must be equal to the provided value.</li> <li>• <b>wildcard</b> - the extracted value must match. This value allows using the asterisk sign (*) (zero or more characters) and the question mark (?) (zero or one character).</li> <li>• <b>regexp</b> - the extracted value must match the provided regular expression.</li> </ul>

### Defining grouping name expression pattern

The pattern element of the expression condition is used to define the grouping name expression pattern to get the value of the **CI** field. The extracted value is used for matching with the value defined in the match element of the expression condition. This pattern can also perform a regular expression matches on the CIs instead of just providing static text names. The grouping name expression pattern supports the limited Apache Velocity syntax, similar to the one used for groupId in the grouping patterns. The only difference is the variable name, which represents the configuration item, which is \$CI.

The two following predefined variables can be used in the grouping name expression pattern for the tiers configuration:

- **\$CI**

This variable represents the core configuration item that is being processed by BizAppsAgent. It can be used to extract attributes of the CI. For example, if the CI is ComputerSystem, the \$CI.OSRunning.OSName grouping name expression pattern extracts the OSName field from OperatingSystem (OSRunning) that runs on a particular ComputerSystem.

- **\$utils**

This variable represents the utilities available in grouping name expression patterns. For more information, see [“Grouping name expression” on page 195](#).

### Attaching a custom configuration to a grouping pattern

You can create your own customized grouping pattern configuration and attach it to grouping patterns.

Grouping patterns configuration controls how business applications are built. You can create a customized configuration, when you want to change the default process. For example, you can exclude certain relations and set the maxHopsLimit property to a lower value to reduce the size of your application. You can create your own tiers, if you are integrating TADDM with other products. See [“Grouping patterns configuration” on page 200](#) for detailed information about each section of the grouping pattern configuration.

**Important:** The only section of the configuration that you cannot customize for a specific grouping pattern is the direction configuration. This section always applies to all grouping patterns, so when you modify it, all patterns are affected.

To create a customized configuration, and attach it to a grouping pattern, complete the following steps.

1. Create an XML configuration file. The following approaches are possible:

- You can create an empty file, save it as the XML file, for example, `my_config.xml`, and provide the content.
- You can export the default configuration to a new file, and modify the existing content. Run the `bizappscli` tool from the `<taddm_installation_directory>dist/sdk/bin` directory in the following way:

```
bizappscli.sh exportDefaultConfiguration -f file name
```

The `-f` option defines the destination file where the default configuration is exported. The following example shows how to export the default configuration to the `my_config.xml` file:

```
bizappscli.sh exportDefaultConfiguration -f my_config.xml
```

**Note:** The examples in this topic are valid for Linux and UNIX operating systems. If you are using Windows operating system, run the tool in the `bizappscli.bat` format.

2. Depending on your needs, provide the custom content of the file that you created or exported.
3. Import the new configuration to the database by running the `bizappscli` tool in the following way:

```
bizappscli.sh importConfiguration -c configuration name -f file name
```

The `-c` option specifies the name of the new configuration, which you can then select from the list of all configurations in Data Management Portal. The `-f` option defines a source file from which the configuration is imported. The following example shows how to import the `custom_config` configuration from the `my_config.xml` file:

```
bizappscli.sh importConfiguration -c custom_config -f my_config.xml
```

4. Attach the configuration a grouping pattern. You can use either the `bizappscli` tool, or Data Management Portal.

- **bizappscli tool**

To attach a configuration to a grouping pattern by using the `bizappscli` tool, run the tool in the following way:

```
bizappscli.sh attachConfiguration -c configuration name -n pattern  
name | -g pattern GUID
```

Use either the `-n`, or `-g` option. The `-n` option defines the name of the grouping pattern. The `-g` option defines the GUID of the grouping pattern.

The following example shows how to attach the `custom_config` configuration to a grouping pattern named `my_pattern`:

```
bizappscli.sh attachConfiguration -c custom_config -n my_pattern
```

**Note:** You can attach a configuration to only one pattern at a time. If you want to attach the same configuration to more patterns, repeat the procedure.

- **Data Management Portal**

To attach a configuration to a grouping pattern by using Data Management Portal, complete the following steps:

- a. Open Data Management Portal.
- b. In the Functions pane, click **Discovery** if you use TADDM 7.3.0, or 7.3.0.1, or **Analytics**, if you use TADDM 7.3.0.2, or later.
- c. Click **Grouping Patterns**.
- d. Select the pattern to which you want to assign the configuration, and click **Edit**.
- e. From the **Configuration** list, select the configuration that you created, for example **custom\_config**.

f. Click **OK**.

If you want the default configuration to be attached back to the pattern that you customized, you must detach the customized configuration. Run the `bizappscli` tool in the following way:

```
bizappscli.sh detachConfiguration -n pattern name | -g pattern GUID
```

Use either the `-n`, or `-g` option. The `-n` option defines the name of the grouping pattern. The `-g` option defines the GUID of the grouping pattern.

The following example shows how to detach the customized configuration from a grouping pattern named `my_pattern`:

```
bizappscli.sh detachConfiguration -n my_pattern
```

### Related tasks

[“Configuring a grouping pattern configuration” on page 241](#)

Learn how to customize a business application by configuring tiers, reversing relations, and importing configurations by using the `bizappscli` tool.

### Related reference

[“Actions for managing grouping pattern configuration” on page 225](#)

By using the `bizappscli` tool, you can export and import whole grouping pattern configurations, or their specific sections.

## Traversing relations during pattern processing

All elements in a business application are traversed and aligned according to the dependency direction between objects. You can view the direction of traversal on a topology.

The objects that are located in the topology above other objects always depend on them. For example, application server is always located above a hosting computer system. It means that the application server depends on the computer system. The direction of a relation that connects these two CIs is represented on the topology by an arrow that points from the application server to the computer system.

For each selector, you can decide which dependent objects to traverse. First, choose from the following two options:

- Traverse dependencies down (LowerDown). When you select this option, BizAppsAgent traverses all objects, on which a specific core CI depends, and all other objects on which these dependent objects depend. For example, BizAppsAgent traverses from the web server to a hosting computer system, on which the web server depends, and then goes from the computer system to a hosting hypervisor, on which the computer system depends.
- Traverse dependencies up (HigherUp). When you select this option, BizAppsAgent traverses all objects which depend on a specific core CI, and all other objects that depend on these objects. For example, BizAppsAgent traverses from a hypervisor to all computer systems that are hosted on it, and then to all application servers that are hosted on these computer systems.

With these two options, you can traverse relations in only one strictly defined direction, which is either up or down. However, when you move one level up or down from the core CI, you can also include processing relations in the opposite direction. For example, BizAppsAgent can start from a specific computer system, go down to its hosting hypervisor, and then go up to all other hosted virtual computer systems. To enable processing relations in the opposite direction, use the following options:

- HigherDown. You can select this option only when the HigherUp option is selected. When you select the HigherDown option, BizAppsAgent goes down from all objects that it encounters while it traverses up the dependencies.
- LowerUp. You can select this option only when the LowerDown option is selected. When you select the LowerUp option, BizAppsAgent goes up from all objects that it encounters while it traverses down the dependencies.





## LowerDown

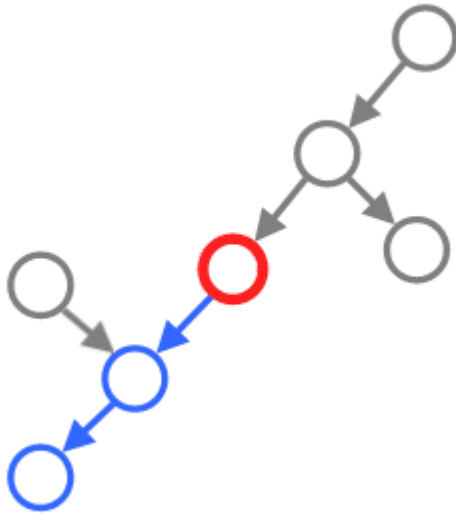


Figure 4. Topology with the LowerDown option selected.

## LowerDown and LowerUp



Figure 5. Topology with the LowerDown and LowerUp options selected.

## Example

The following diagrams show an example business application with various traversal options selected. The red circle represents a core CI, and the blue circles represent objects that are included in the business application.

### Traversing dependencies with the HigherUp option

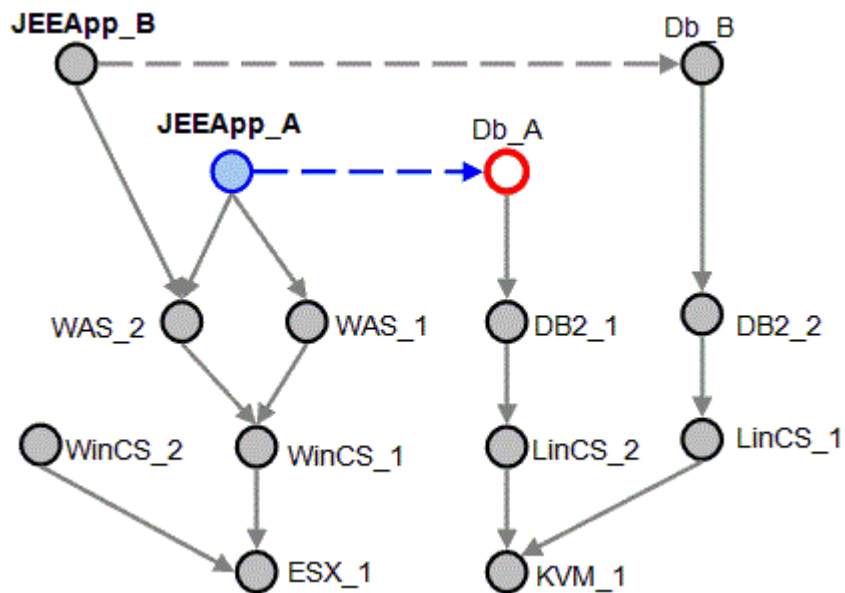


Figure 6. Topology with only HigherUp option selected.

### Traversing dependencies with HigherUp and HigherDown options

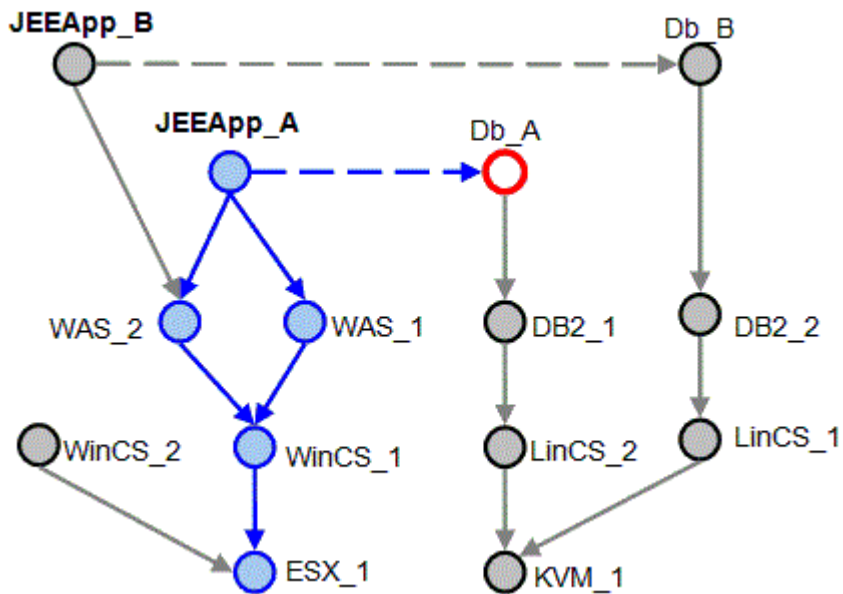


Figure 7. Topology with HigherUp and HigherDown options selected.

### Traversing dependencies with the LowerDown option

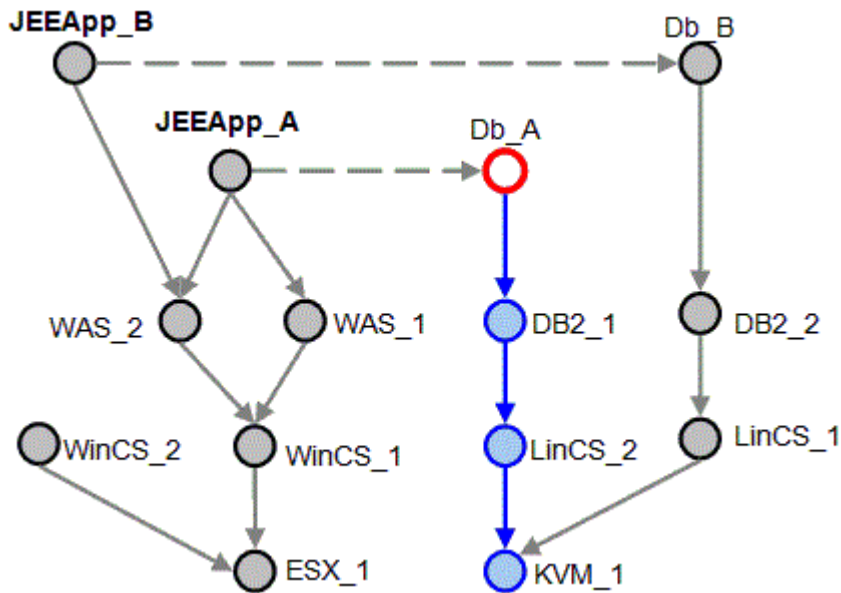


Figure 8. Topology with only LowerDown option selected.

### Traversing dependencies with LowerDown and LowerUp options

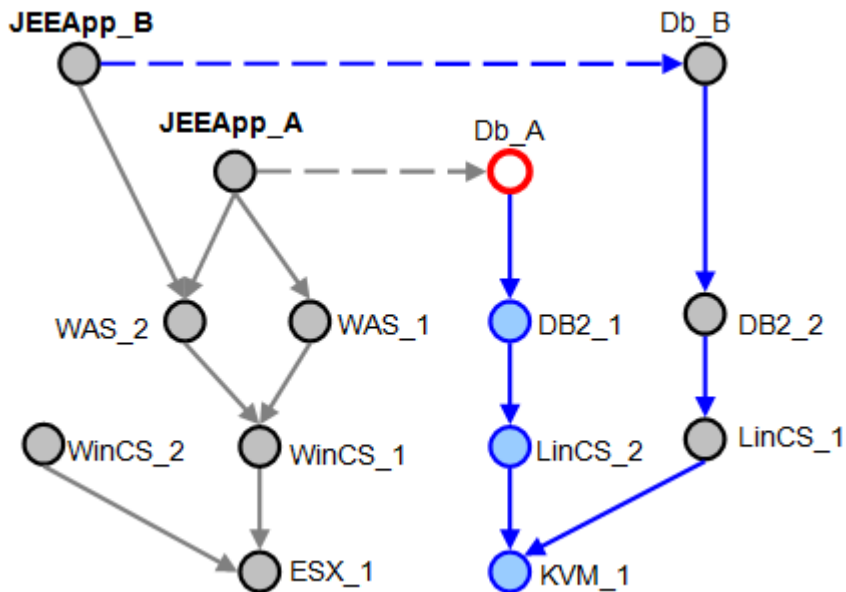


Figure 9. Topology with LowerDown and LowerUp options selected.

## bizappscli tool

You can use the CLI `bizappscli` tool to manage grouping patterns, grouping pattern processing schedules, grouping pattern configurations and the execution of grouping patterns.

**Fix Pack 2** You can use the tool to create reports for analyzing the content of business applications.

**Fix Pack 3** You can use the tool to export the graph of the business application topology to the SVG format.

The script is in the `<taddm installation directory>/dist/sdk/bin` directory. Depending on the operating system, the following formats of the script are available:

- For Linux, AIX, and Linux on System z - `bizappscli.sh`.
- For Windows - `bizappscli.bat`.

**Note:** In the bizappsccli tool section, bizappsccli.sh format is used in all examples. If you work on Windows operating system, remember to use bizappsccli.bat format.

**Important:** Fix Pack 3 To use the bizappsccli tool, you must have the Update permission granted for the DefaultAccessCollection.

To run the bizappsccli tool, you must specify an action and options. The following options apply to all actions:

- -H, --hostname <arg> - defines the host name. The default value is 0.0.0.0. If you use the -T parameter, you must also specify the -H parameter.
- -P, --port <arg> - defines the port. The default value is 9433.
- -p, --password <arg> - defines the password of the TADDM user.
- -T, --truststore <arg> - defines the location of the truststore file, jssecacerts.cert, with a certificate for connection to the TADDM server. This parameter is required for secure connection to TADDM. If you use this parameter, you must also specify the -H parameter.
- -u, --user <arg> - defines the TADDM user name.

**Note:** When the values of the options contain spaces, you must enclose the values in double quotation marks, for example "my grouping pattern". Otherwise, the tool interprets a word after the first space as the name of the action, and an error is generated.

To see all available actions, run the script without any arguments. To see a description of an action, use the following script:

```
bizappsccli.sh help -a <action>
```

For example, bizappsccli.sh help -a listPatterns.

You can use the bizappsccli.sh tool to control the processing of grouping patterns. Multiple patterns can be processed on a single storage server, which by default is primary storage server, and also on multiple storage servers.

### Execution groups

The patterns are processed inside a thread pool called ExecutionGroup. Each storage server can devote threads to a particular execution group, which becomes a part of multiple servers thread pool, which then processes patterns. The size of the thread pool defines how many patterns can be run in parallel. Each pattern can be processed by five threads at the same time, which is controlled by the ExecutionGroup mechanism.

If at a given time there are more grouping patterns that require processing than the capacity of the execution group allows for, the outstanding patterns are run at the nearest possible time after one of the threads of the execution group becomes available. A new execution group is created when there is at least one schedule that refers to the execution group name. However, when the execution group is created, it must be enabled on each storage server that is to be a part of that execution group, and the thread pool size needs to be configured separately on each storage server.

By default, there is one execution group available, called default. The group is configured to have only one thread running only on the primary storage server (it does not need to be enabled on the primary server). Other execution groups can be created to make sure that the higher priority patterns always have available threads. All patterns that run inside execution groups have the same priority, and therefore one pattern can block the processing of another if no idle threads are available within the group. The creation of another execution group constitutes the creation of another thread pool.

You can enable the processing of the execution groups on secondary storage servers in addition to the primary storage server. On each secondary storage server, set the following property in the collation.properties file to true:

```
com.ibm.cdb.internalscheduling.bizapps.<GROUP NAME>.enabled=true
```

<GROUP NAME> is the name of the group that you want to enable. For example, to enable the default group, set the following property:

```
com.ibm.cdb.internalscheduling.bizapps.default.enabled=true
```

## Execution schedules

Patterns are run according to their schedule. Each pattern must be associated with a schedule during the creation. A schedule object contains information about when, or how often, the pattern is processed, and what execution group is used for that processing. By default, only one schedule called default is created, with a default interval set to every 4 hours. This default schedule is associated with the default execution group.

If you migrated business applications from TADDM 7.2.2, the default interval is the same as the value of the `com.ibm.cdb.topobuilder.groupinterval.bizapps` property. You can check the value of this property in the `collation.properties` file.

Schedules can be based either on intervals or on the cron expression.

### Interval

An interval-based schedule triggers the pattern processing periodically with defined intervals. The first processing is triggered after the first interval time passes after the TADDM server is started.

### Cron

A cron-based schedule can be created by specifying the cron expression, which allows for more complex schedules, such as 'every Thursday at 18:00'.

For more information about intervals, and cron expressions, see [“Actions for managing grouping pattern schedules”](#) on page 222.

## Single pattern manual testing

You can test a single pattern manually outside of the defined schedule.

A manual run always triggers a run in the execution group according to the defined schedule. As execution group threads might be spread across multiple servers, the thread that is used for processing might be on a server other than the one used for starting the processing.

The manual run does not change the defined schedule that is associated with the pattern. However, as any pattern can be processed only once at the same time, if a pattern is running because it was triggered by a defined schedule, the manual run fails. When pattern processing was interrupted, the next processing is started according to the defined schedule.

When a pattern is run manually, and an automatic run is scheduled at the same time, the scheduled run is run as soon as possible. It means that the scheduled run is run when the manual run finishes and there is a free thread available in the associated execution group.

## Configuring bizappsccli tool properties

### com.ibm.taddm.bizappsccli.jvmArgs=-Xmx1024M

This property defines JVM options for the `bizappsccli` tool usage. If there is not enough memory to run the tool, you can increase the maximum heap size of JVM processes by modifying the value of this property in the `$COLLATION_HOME/sdk/etc/collation.properties` file. For example, exporting business application topologies to the SVG format might require more memory than defined in the default settings.

Set the value in the following format: `-Xmx1024M`. In this example, the maximum heap size is 1024 MB (1 GB).

## Actions for managing grouping patterns

By using the `bizappsccli` tool, you can export and import grouping patterns and control how they are run.

You can use the following actions, for example, when you copy grouping patterns between development, test, or production environments.

## exportPatterns

This action exports grouping pattern definitions. You can import them later to the same or another TADDM instance. You can separately customize the export mode of the configuration (the `-C` option) and schedule (the `-S` option). The following values are available:

- NEVER - the associated configuration or schedule is never exported.
- ALWAYS - the associated configuration or schedule is always exported.
- NONDEFAULT - the associated configuration or schedule is exported if it is not the default one.

The following options are available:

- `-C, --configurationExportMode <arg>` - defines the export mode of the configuration. The default value is NONDEFAULT.
- `-f, --filename <arg>` - defines the destination file where patterns are exported. If you do not specify a path, the file is created in the current working directory. If you do not specify this file, the results are printed to the standard output.
- `-g, --guid <arg>` - defines the GUID of the pattern.
- `-n, --name <arg>` - defines the name of the pattern.
- `-S, --scheduleExportMode <arg>` - defines the export mode of the schedule. The default value is NONDEFAULT.

For example, to export a grouping pattern with the name `gp2` to a file `new_pattern.xml` but without a schedule, run the tool in the following way:

```
bizappscli.sh exportPatterns -n gp2 -f new_pattern.xml -S NEVER
```

## importPatterns

This action loads grouping pattern definitions which were exported to an XML file. If required, the definitions contain defined schedules and configuration settings. You can customize the import mode of the pattern (the `-I` option), schedule (the `-S` option), and configuration (the `-C` option). The following list contains the available values.

**Note:** In the following description of the available values, the object stands for a pattern, schedule, or configuration, depending on which option you use.

- DEFAULT - for schedule and configuration, this option attaches the pattern to the default object. Imported data is skipped.
- ATTACH - looks for the existing objects by name. If the object is found, the pattern, schedule, or configuration is attached to it. If the object is not found, the command fails. The imported data is skipped.
- CREATE - creates an object. If an object with the same name exists, the command fails.
- UPDATE - looks for the existing objects by name and updates them with the imported data. If an object does not exist, it is created.

The following options are available:

- `-C, --configurationImportMode <arg>` - defines the import mode of a configuration. The default value is CREATE.
- `-f, --filename <arg>` - defines the source file from which patterns are imported.
- `-I, --patternImportMode <arg>` - defines the import mode of a pattern. The default value is CREATE.
- `-r, --filter <arg>` - selects the patterns that are imported from the source file. The value of this option is the name of the grouping pattern. You can optionally use a question mark (?) and an asterisk (\*) as a wildcard. For example, to import patterns that have names which start with the letter "c", specify the `c*` value.

- `-S, --scheduleImportMode <arg>` - defines the import mode of the schedule. The default value is `CREATE`.
- `-X, --prefix <arg>` - defines an optional phrase that you can use to prefix the names of the imported patterns, schedules, and configurations.

**Important:** The `-f` option is required.

For example, to import a pattern that you saved in the `my_pattern.xml` file, add to its name the prefix `10_Jun`, and make sure that the default schedule is assigned to it, run the tool in the following way:

```
bizappscli.sh importPatterns -f my_pattern.xml -X 10_Jun -S DEFAULT
```

## restoreExamplePatterns

This action re-creates the default example pattern definitions that are available after TADDM installation. You can use the following import mode options to specify how to import example patterns:

- `-C, --configurationImportMode <arg>` - defines the import mode of the configuration. The default value is `CREATE`.
- `-I, --patternImportMode <arg>` - defines the import mode of the pattern. The default value is `CREATE`.
- `-S, --scheduleImportMode <arg>` - defines the import mode of the schedule. The default value is `CREATE`.
- `-X, --prefix <arg>` - defines an optional phrase that you can use to prefix the names of the imported patterns, schedules, and configurations.

For example, to restore the default example grouping patterns and make sure that the default configuration is assigned to them, run the tool in the following way:

```
bizappscli.sh restoreExamplePatterns -C DEFAULT
```

## listPatterns

This action lists the grouping patterns and general information about them. By default, it displays execution group ID, schedule name, grouping pattern name, and the time of the next execution of the pattern.

The following options are available:

- `-A, --sort` - sorts the results by execution groups and schedules.
- `-C, --showConfig` - shows configurations of the patterns.
- `-F, --dateFormat <arg>` - specifies the date format. The default format is `M/d/yy h:mm a`, for example `03/25/15 2:38 PM`. For other date formats, look for information about the `SimpleDateFormat` Java class.
- `-g, --guid <arg>` - limits the results to one specific pattern.
- `-G, --showGuids` - shows the GUIDs of the patterns.
- `-N, --hideNames` - hides names of the patterns. This option is typically used with the `-G` option.
- `-S, --separator <arg>` - defines the separator of the values. The default separator is a space, " ".

For example, to display information about only one pattern with the GUID `00000000000000000000000000000000` and add a time zone information to the time of the next execution of the pattern, run the tool in the following way:

```
bizappscli.sh listPatterns -g 00000000000000000000000000000000 -F "M/d/yy h:mm a z"
```

Sample output:



EXECUTION_GROUP	SCHEDULE_NAME	PATTERN_NAME	NEXT_EXECUTION_TIME
default	schedule1	my_pattern1	7/16/15 3:38 AM GMT-12:00

## showPatternsRunning

This action lists the jobs that are currently running on all schedulers and the instances of these schedulers. By default, only the name of the grouping pattern that is running is displayed. The following options are available:

- -G, --showGuids - shows the GUIDs of the patterns.
- -I, --showServerID - shows the processing server.
- -N, --hideNames - hides names of the patterns. This option is typically used with the -G option.
- -S, --separator <arg> - defines the separator of the values. The default separator is a space, " ".

For example, the following command displays the GUIDs, and the server which processes the grouping patterns. Additionally, the values are separated by semicolon (;) to distinguish the values that have spaces in names.

```
bizappsccli.sh showPatternsRunning -G -I -S ;
```

Sample output:

```
00000000000000000000000000000000;my pattern 1;taddmserverA.ibm.com
11111111111111111111111111111111;my pattern 2;taddmserverB.ibm.com
22222222222222222222222222222222;my pattern 3;taddmserverA.ibm.com
```

## runPattern

This action schedules a pattern to be run immediately when a thread is available. The following options are available:

- -e, --executionGroupId <arg> - defines the name of the execution group from which the patterns are run.
- -g, --guid <arg> - defines the GUID of the pattern.
- -l, --infoLevel <arg> - defines the details level of the log files. The following values are available:
  - NOINFO - no information.
  - DEFAULT - information about starting and stopping the business application only.
  - GENERAL - the DEFAULT information and core CI data.
  - DETAILS - the GENERAL information and composition engine traversal route details.
  - MAXINFO - maximum detail level, currently the same as for the DETAILS level.

If you do not specify any value, the details level is the same as in the configuration of the grouping pattern that is run.

- -n, --name <arg> - defines the name of the pattern.
- -w, --wait - defines whether to wait until the pattern finishes running. The default value of this option is false.

**Important:** One of the following options is required: -e, -g, or -n.

For example, to schedule the pattern with the name my pattern 5 to run when a thread is available, and set the details of the log file to the highest level, run the tool in the following way:

```
bizappsccli.sh runPattern -n "my pattern 5" -l MAXINFO
```

## stopPatternRun

This action stops the currently running job. The following options are available:

- -g, --guid <arg> - defines the GUID of the pattern that you want to stop.
- -n, --name <arg> - defines the name of the pattern that you want to stop.

**Important:** One of the following options is required: -g, or -n.

For example, to stop a pattern with the GUID 00000000000000000000000000000000, run the tool in the following way:

```
bizappsccli.sh stopPatternRun -g 00000000000000000000000000000000
```

## deletePatterns

This action removes one or more patterns. Additionally, all future scheduled executions are cleared. The following options are available:

- -f, --filter <arg> - defines the names of the patterns to remove. You can optionally use a question mark (?) and an asterisk (\*) as a wildcard. For example, to import patterns that have names which start with the letter "c", specify the c\* value.
- -g, --guid <arg> - defines the GUID of the pattern that you want to remove.

**Important:** One of the following options is required: -f, or -g.

For example, to remove all grouping patterns that have names which start with the letter "p", run the tool in the following way:

```
bizappsccli.sh deletePattern -f p*
```

## Actions for managing grouping pattern schedules

By using the bizappsccli tool, you can create and modify grouping patterns schedules.

### createSchedule

This action creates schedules. The following options are available:

- -c, --cronExpression<arg> - defines the interval by using the cron expression. For more information about the cron expression format, see the Quartz Scheduler documentation at <http://quartz-scheduler.org/documentation/quartz-2.x/tutorials/tutorial-lesson-06>.

**Note:** The cron expressions are not compatible with the format that is used by cron daemon on Linux and other UNIX operating systems. There is an additional sixth mandatory field to represent seconds.

- -d, --description <arg> - defines the description of the schedule.
- -e, --executionGroupId <arg> - defines the execution group ID. The default execution group ID is default.
- -i, --interval <arg> - defines the execution interval in hours.
- -l, --misfireLimit <arg> - defines how long a pattern can wait to be run before it is skipped from processing and a warning is logged. The warning contains the information that the pattern did not start. To prevent it, you can create more thread pools, or modify the schedules of your patterns. The value of this option is expressed in minutes. The default value is 60. If you do not want to modify your configuration and at the same time you do not want the patterns to be skipped, set this option to a higher value. You can also modify this limit by using the `com.ibm.cdb.internalscheduling.bizapps.schedule name.misfire` property in the `collation.properties` file.
- -n, --name <arg> - defines the name of the schedule. It is possible to create two or more schedules with the same name. However, it is not advised because it might lead to confusion.

**Important:** The -n option is required, and you must also use either the -c, or the -i option.

Examples:

- To create a schedule that is named every three hours and that runs at the interval of 3 hours, run the tool in the following way:

```
bizappsccli.sh createSchedule -n "every three hours" -i 3
```

- To create a schedule that is named every midnight and that runs every day at midnight, run the tool in the following way:

```
bizappsccli.sh createSchedule -n "every midnight" -c "0 0 0 * * ?"
```

- To create a schedule that is named Saturdays and that runs every Saturday at 8 p.m., run the tool in the following way. The -d option adds a description to clarify the usage of the schedule.

```
bizappsccli.sh createSchedule -n "Saturdays" -c "0 0 20 ? * SAT" -d "every Saturday at 8 PM"
```

## listSchedules

This action lists all schedules. By default, it displays execution group ID, schedule name, the interval at which the schedule is run, and the description of the schedule if it is provided.

The following options are available:

- -G, --showGuids - displays GUIDs of the schedules.
- -N, --hideNames - hides names of the schedules. This option is typically used with the -G option.
- -S, --separator <arg> - defines the separator of the values. The default separator is a space, " ".

For example, to list all schedules with their GUIDs, and separate the values with the forward slash (/), run the tool in the following way:

```
bizappsccli.sh listSchedules -G -S /
```

Sample output:

```
GUID EXECUTION_GROUP NAME INTERVAL DESCRIPTION
00000000000000000000000000000000/default/every three hours/INTERVAL: 3h
11111111111111111111111111111111/default/default/INTERVAL: 4h
22222222222222222222222222222222/default/Saturdays/CRON: 0 0 20 ? * SAT/
"every Saturday at 8 PM"
33333333333333333333333333333333/default/every midnight/CRON: 1 1 1 * * ?
```

## updateSchedule

This action updates the schedule configuration. The following options are available:

- -c, --cronExpression <arg> - defines the new interval by using the cron expression.
- -d, --description <arg> - defines the new schedule description.
- -e, --executionGroupId <arg> - defines the new execution group ID. The default execution group ID is default.
- -g, --guid <arg> - defines the GUID of the schedule that you want to update.
- -i, --interval <arg> - defines the new execution interval in hours.
- -l, --misfireLimit <arg> - defines how long a pattern can wait to be run before a warning is logged. The warning contains information that the pattern has not started. To prevent these warnings to be logged, you might create more thread pools, or modify the schedules of your patterns. The value of this option is expressed in minutes. If you do not want to modify your configuration and at the same time you do not want the warnings to be logged, set this option to a higher value.
- -n, --name <arg> - defines the new name of the schedule.

**Important:** The -g option is required.

Examples:

- To rename a schedule with the GUID 00000000000000000000000000000000 to new\_schedule, run the tool in the following way:

```
bizappscli.sh updateSchedule -g 00000000000000000000000000000000 -n  
"new schedule"
```

- To disable the schedule with the GUID 00000000000000000000000000000000, specify the OFF value for the -i option. The -d option adds a description to clarify the usage of the schedule.

```
bizappscli.sh updateSchedule -g 00000000000000000000000000000000 -i OFF -d  
"on demand only"
```

## changeDefaultSchedule

This action modifies the default schedule without specifying the GUID. The following options are available:

- -c, --cronExpression <arg> - defines the new interval by using the cron expression.
- -i, --interval <arg> - defines the new interval in hours.

**Important:** One of the following options is required: -c, or -i.

For example, to change the default schedule to run every day at midnight instead of every 4 hours, run the tool in the following way:

```
bizappscli.sh changeDefaultSchedule -c "0 0 0 * * ?"
```

## deleteSchedule

This action deletes a schedule with a specific GUID. You cannot remove the default schedule. The following options are available:

- -f, --forceToDefault - forces the schedule to be removed even if it is assigned to a grouping pattern. The default schedule is used instead. If you do not use this option and want to remove a schedule that is assigned to a grouping pattern, an error occurs and the schedule is not removed.
- -g, --guid <arg> - defines the GUID of the schedule that you want to remove.

**Important:** The -g option is required.

For example, to delete a schedule with a GUID 00000000000000000000000000000000 even if it is assigned to a grouping pattern, run the tool in the following way:

```
bizappscli.sh deleteSchedule -f -g 00000000000000000000000000000000
```

## resetSchedules

This action resets all grouping pattern schedules. It means that when there is a certain number of jobs to be processed, and some of them are started and some are queued, all of the jobs are restarted. The grouping patterns that are already being processed are not stopped, but the patterns that are queued are not started. When schedules are set to run at intervals, the intervals are restarted. For example, if an interval is set to 3 hours, and you restart it, the schedule is run 3 hours after you ran the resetSchedules action. When a schedule has a cron expression set, nothing changes. You might need it when direct SQL changes are made to grouping patterns.

This action has no additional options.

To reset all schedules, run the tool in the following way:

```
bizappscli.sh resetSchedules
```

## Actions for managing grouping pattern configuration

By using the `bizappsccli` tool, you can export and import whole grouping pattern configurations, or their specific sections.

### exportConfiguration

This action exports a grouping pattern configuration. The following options are available:

- `-c, --config <arg>` - defines the name of the configuration that you want to export.
- `-f, --filename <arg>` - defines the destination file where the configuration is exported. If you do not specify a path, the file is created in the current working directory. If you do not specify the file name, the results are printed to the standard output.
- `-l, --general` - exports the general section of the configuration.
- `-m, --composition` - exports the composition section of the configuration.
- `-t, --tiers` - exports the tiers section of the configuration.
- `-v, --traversal` - exports the traversal section of the configuration.

**Important:** The `-c` option is required.

For example, to export the composition and traversal sections of the configuration with the name `config2` into the `composition_traversal.xml` file, run the tool in the following way:

```
bizappsccli.sh exportConfiguration -c config2 -f composition_traversal.xml -m -v
```

### importConfiguration

This action imports a grouping pattern configuration. The following options are available:

- `-c, --config <arg>` - defines the name of the configuration that you want to import.
- `-f, --filename <arg>` - defines the source file from which the configuration is imported.
- `-U, --allowUpdate` - updates the existing configuration if you are importing a modified file instead of a new one.

**Important:** The `-c` and `-f` options are required.

For example, to import the configuration with the name `custom1` from the `db_tiers.xml` file, run the tool in the following way:

```
bizappsccli.sh importConfiguration -c custom1 -f db_tiers.xml
```

### exportDefaultConfiguration

This action exports the default grouping pattern configuration. By the default, the general, composition, tiers and traversal sections of the default configuration are exported, the direction section is skipped. The following options are available:

- `-d, --directions` - exports the direction section of the configuration.

**Important:** If you want to export the default configuration to edit it and use it as a customized configuration for a specific grouping pattern, do not use the `-d` option. The direction configuration always applies to all grouping patterns and you cannot customize it for a specific grouping pattern.

- `-f, --filename <arg>` - defines the destination file where the default configuration is exported. If you do not specify a path, the file is created in the current working directory. If you do not specify the file name, the results are printed to the standard output.
- `-l, --general` - exports the general section of the configuration.
- `-m, --composition` - exports the composition section of the configuration.
- `-t, --tiers` - exports the tiers section of the configuration.

- `-v, --traversal` - exports the traversal section of the configuration.

For example, to export the directions and general sections of the default configuration into the `dir_gen.xml` file, run the tool in the following way:

```
bizappscli.sh exportDefaultConfiguration -d -l -f dir_gen.xml
```

## importDefaultConfiguration

This action imports the default grouping pattern configuration. The following option is available:

- `-f, --filename <arg>` - defines the source file from which the default configuration is imported.

**Important:** The `-f` option is required.

For example, to import the default configuration from the `my_default_config.xml` file, run the tool in the following way:

```
bizappscli.sh importDefaultConfiguration -f my_default_config.xml
```

## deleteConfiguration

This action removes a grouping pattern configuration. The following option is available:

- `-c --config <arg>` - defines the name of the configuration that you want to delete.

**Important:** The `-c` option is required.

For example, to delete a configuration with the name `test_config`, run the tool in the following way:

```
bizappscli.sh deleteConfiguration -c test_config
```

## attachConfiguration

This action assigns the existing grouping pattern configuration to a specific grouping pattern. The following options are available:

- `-c, --config <arg>` - defines the name of the configuration.
- `-g, --guid <arg>` - defines the GUID of the grouping pattern.
- `-n, --patternname <arg>` - defines the name of the grouping pattern.

**Important:** The `-c` option is required, and you must also use either the `-g`, or the `-n` option.

For example, to attach a configuration with the name `custom_config` to a grouping pattern named `pattern1`, run the tool in the following way:

```
bizappscli.sh attachConfiguration -c custom_config -n pattern1
```

**Note:** You cannot use this action to attach the default configuration to a pattern for which you created a customized configuration. If you want to go back to the default configuration, you must detach the customized configuration from the grouping pattern. See the next section.

## detachConfiguration

This action removes a customized configuration from a grouping pattern. The following options are available:

- `-g, --guid <arg>` - defines the GUID of the grouping pattern.
- `-n, --patternname <arg>` - defines the name of the grouping pattern.

**Important:** One of the following options is required: `-g`, or `-n`.

For example, to detach a customized configuration from a grouping pattern with GUID 00000000000000000000000000000000, run the tool in the following way:

```
bizappsccli.sh detachConfiguration -g 00000000000000000000000000000000
```

## listConfigurations

This action lists all grouping pattern configurations. By default, it displays only configuration names.

The following options are available:

- -G, --showGuids - shows the GUIDs of the configurations.
- -n, --patternGuid <arg> - shows the configuration that is attached to a specific grouping pattern.
- -s, --showPatterns - lists all configurations with all grouping patterns that are currently attached to them.

For example, to generate a list of configurations and grouping patterns that are attached to them, run the tool in the following way:

```
bizappsccli.sh listConfigurations -s
```

Sample output:

```
Configuration: custom1
Attached patterns:
pattern1
pattern2

Configuration: config2
Attached patterns:
gp1
gp2
```

## Actions for managing execution groups

By using the bizappsccli tool, you can display execution groups.

### showExecutionGroups

This action lists all execution groups. By default, the list contains the names of the groups and the number of configured threads that are assigned to each group. The following options are available:

- -D, --showServerDetails - provides detailed information about each server.
- -S, --separator <arg> - defines the separator of the values. The default separator is a space, " ".

For example, to generate a list of execution groups, display information about the processing servers, and separate the values with a semicolon (;), run the tool in the following way:

```
bizappsccli.sh showExecutionGroups -D -S ;
```

Sample output:

```
GROUP: default;          TOTAL THREADS: 5;
NODES:
NODE: vmw000000000000.taddmserverA.ibm.com;  THREADS: 5
```

## Configuring execution groups for each server

If you want to configure execution groups separately for each server, use the following properties in the collation.properties file.

### com.ibm.cdb.internalscheduling.bizapps.<GROUP NAME>.enabled

Determines whether to enable or disable the provided execution group on the server.

The allowed values are true and false.

**com.ibm.cdb.internalscheduling.bizapps.<GROUP NAME>.threads**

Specifies the number of available threads for the provided execution group.

The value is specified in numbers, for example 5.

## **Fix Pack 2 Actions for analyzing the content of business applications**

By using the `bizappsccli` tool, you can generate reports, which you can use to analyze the content of your business applications.

### **analytics**

This action creates reports, which you can then use in the analysis of your business applications content and their creation process. You can display one report at a time. The following reports are available:

- -a, --AbandonedCollections - displays the names of these applications, for which a pattern no longer exists. The pattern was deleted, or it was modified in such a way that it no longer generates application with the same ID. Topologies of such business applications are not updated anymore. The following information is displayed:
  - COLLECTION\_NAME
  - LAST\_GENERATION\_TIME
- -b, --PlaceholdersByProc - displays already discovered placeholder servers that are a part of at least one business application and that are matched with unrecognized processes. This report is useful when you create new custom server templates for the processes which are not recognized by default. The following information is displayed:
  - MATCHED\_PROCESS
  - PLACEHOLDERS\_COUNT
- -c, --Counts - displays the number of nodes and paths of business applications. This report helps you quickly detect anomalies such as too large or too small applications. The following information is displayed:
  - COLLECTION\_NAME
  - NODES\_COUNT
  - PATHS\_COUNT
  - LAST\_GENERATION\_TIME
- -e, --ForRegeneration - displays the applications in which member nodes refer to the nodes that were deleted after the last generation of application. Such applications contain outdated content and must be regenerated. The following information is displayed:
  - COLLECTION\_NAME
  - NODES\_COUNT
- -h, --AllPlaceholders - displays all placeholder servers that are a part of business applications. This report provides detailed information about each placeholder server. The following information is displayed:
  - PLACEHOLDER
  - MATCHED\_PROCESS
  - HOST
  - LAST\_STORE\_TIME
  - COLLECTION\_NAME
- -n, --PlaceholdersNeverDiscovered - displays the placeholder servers that are a part of at least one business application and that do not contain any host information. Typically, such servers are a part



of an environment that was not yet discovered, or an external environment, for example cloud. The following information is displayed:

- PLACEHOLDER
- COLLECTION\_NAME
- -o, --UnusedRoutes - displays the routes that were traversed but were not used in building paths. The relations that are represented by these routes might be excluded from traversing in the grouping pattern configuration. Analyze carefully the routes that are unused, because they might be used again when new data is introduced. The following information is displayed:
  - RELATION\_TYPE
  - COUNT
- -r, --Routes - displays routes that are used to generate business applications. Routes are not displayed on the topology, but they build paths that are displayed on the topology. By adding the -g option, you can display the routes for a particular application. The following information is displayed:
  - RELATION\_TYPE
  - SOURCE\_TYPE
  - TARGET\_TYPE
  - COUNT
- -w, --PlaceholdersToRediscover - displays the placeholder servers that are a part of at least one business application and that contain hosts, but no matched processes. This means that the placeholder server host is in the scope of the discovery, but the information about unrecognized runtime processes was already deleted. The following information is displayed:
  - PLACEHOLDER
  - HOST
  - LAST\_STORE\_TIME
  - COLLECTION\_NAME

You can use the following options when you create reports with the `analytics` action:

- -A, --sortAsc <arg> - sorts the names of business applications in the alphabetical order. The possible values are `true` and `false`. The default value is `false`.
- -g, --guid <arg> - defines the GUID of an application. You can use it to limit the output of some of the reports to one specific business application.
- -G, --showGuids - shows the GUIDs of the objects, where an object can be an application, placeholder, process, or host.
- -m, --maxRows <arg> - limits the output to the specified number of rows.
- -R, --showPattern - displays the names of grouping patterns. The possible values are `true` and `false`. The default value is `false`.
- -S, --separator <arg> - defines the separator of the values. The default separator is a space, " ".

## Examples

1. The following command displays a report that contains information about the number of nodes and paths of an application with the GUID `00000000000000000000000000000000`:

```
bizappscli.sh analytics -c -g 00000000000000000000000000000000
```

Sample output:

COLLECTION_NAME	COLLECTION_GUID	NODES_COUNT	PATHS_COUNT
LAST_GENERATION_TIME			
pattern_1	00000000000000000000000000000000	2	2 07/09/2015 07:57:07

2. The following command displays a report that contains information about all placeholder servers, and additionally specifies GUIDs of the objects:

```
bizappscri.sh analytics -h -G
```

Sample output:

[illegible]

See also the *Configuring for discovery of placeholders* topic in the TADDM Administrator's Guide.

## Fix Pack 3 Actions for exporting topologies to the SVG format

By using the `bizappscli` tool, you can export your business application topology to the SVG format.

## exportTopology

This action exports the business application topology to the SVG format. The following options are available:

- `-f, --filename <arg>` - defines the destination file where the topology is exported. If you do not specify a path, the file is created in the current working directory. If the file with the specified name exists, it is overwritten. If you do not specify the file name, the results are printed on the standard output.
- `-g, --guid <arg>` - defines the GUID of the business application.
- `-n, --name <arg>` - defines the name of the business application.
- `-z, --zip` - compresses the destination file to the `.zip` format. The `.zip` extension is added automatically. You can use it only with the `-f` option.

**Important:** One of the following options is required: -g, or -n.

For example, to export the topology of the business application named `my_app` to the output file named `my_app_graph.svg`, run the tool in the following way:

```
bizappscli.sh exportTopology -n my_app -f my_app_graph.svg
```

**Limitation:** The business application topology is always exported with highlighted core CIs. If you want the changed CIs, not the core CIs, to be highlighted, follow the procedure that is described in the “Exporting a topology” on page 87 topic.

**Tip:** When the exported topology is displayed, CI labels that are long are truncated. To view the full labels, hover the mouse over a CI node in the graph to display the tooltip, where you can find the full label and GUID. A viewer compliant with SVG 1.1 standard is required, for example Mozilla Firefox web browser.

## Related properties

When you have large business applications, you might need to modify the default values for the following properties:

- `com.ibm.taddm.bizappscli.jvmArgs=-Xmx1024M`  
This property defines JVM options for the `bizappscli` tool usage. If there is not enough memory to run the tool, you can increase the maximum heap size of JVM processes by modifying the value of this property in the `$COLLATION_HOME/sdk/etc/collation.properties` file.
- `com.collation.view.maxnodes=500`  
This property specifies the maximum number of nodes that can be viewed in a topology graph in the Data Management Portal. You can modify the default value in the `$COLLATION_HOME/dist/etc/collation.properties` file.

## Configuring the `collation.properties` file entries

You can configure the following entries related to business applications in the `collation.properties` file.

### **`com.ibm.cdb.serviceinfrastructure.deadlock.retry.count`**

Defines the number of attempts to store a transaction before it fails. When BizAppsAgent is run concurrently in multiple threads, a deadlock on database might occur. To prevent it, attempts to store a transaction are repeated. When the number of attempts exceed the number that is specified by this property, the transaction fails.

The default value is 30.

### **`com.ibm.cdb.serviceinfrastructure.deadlock.retry.time`**

Specifies the timeout before all attempts to store a transaction fail.

The default value is 7200, expressed in seconds (that is 2 hours).

### **`com.ibm.cdb.serviceinfrastructure.path.max.length`**

Defines the maximum length of a business application path, which is the maximum number of segments in one business application path route.

The default value is 9.

Long routes might indicate that data in the database is broken and might lead to an enormous number of queries. In such case, setting the property to a lower value improves the performance of large applications building.

For more information about paths and routes, see [“Business application structure” on page 177](#).

### **`com.ibm.cdb.serviceinfrastructure.afterprocessing.cleansources`**

Determines whether the element sources that are used to build business applications are deleted after the building is finished. If this property is set to TRUE, business applications temporary data is not deleted.

The default value is TRUE.

If you set this property to FALSE, the performance is considerably decreased.

### **`com.ibm.cdb.serviceinfrastructure.sources.transaction.buffer`**

Defines the number of source elements that are stored in one transaction.

The default value is 1000.

If you notice that there are too many deadlocks when large business applications are processed, decrease the value of this property.

### **`com.collation.Tomcat.jvmargs (TADDM 7.3.0 only)`**

Defines JVM options for Data Management Portal. This property can be used to define the maximum heap size. Set this property in the `collation.properties` file, when the default settings are not sufficient to display topologies.

Set the value in the following format: `-Xmx2048M`. In this example, the maximum heap size is 2048 MB (2 GB). You can use any value.

After you change the property, restart the TADDM server.

### **Fix Pack 1 `com.collation.Liberty.jvmargs`**

Defines JVM options for Data Management Portal. This property can be used to define the maximum heap size. Set this property in the `collation.properties` file, when the default settings are not sufficient to display topologies.

Set the value in the following format: `-Xmx2048M`. In this example, the maximum heap size is 2048 MB (2 GB). You can use any value.

After you change the property, restart the TADDM server.

### **`com.collation.topology.maxnodes`**

Defines the maximum number of nodes that can be viewed in a topology. This property defines the topology size more precisely than the default settings. Set this property in the `collation.properties` file, when the default settings are not sufficient to display topologies.

Set the value in the following format: 1000. In this example, the maximum number of nodes is 1000. Values that are too high might lead to Out of Memory errors. If some topologies cause Out of Memory errors, set this property to a lower value.

See also [“Configuring the maximum size of displayed topologies” on page 199](#).

**com.collation.topology.autozoom.threshold**

Defines the minimum size of topology (defined by a number of nodes) that is necessary for a topology to be automatically zoomed in when displayed. Smaller topologies are displayed to fit the view mode. This property improves readability of large topologies.

The default value is 200.

**com.ibm.cdb.serviceinfrastructure.earlier.ver.compatibility**

Determines whether to create objects of the deprecated group types when old business applications are converted by using grouping patterns.

The default value is *TRUE* for the upgrade scenario and *FALSE* for the fresh installation scenario.

For more information, see [“Migration from 7.2.2 and automatic conversion of old business applications” on page 233](#).

**com.ibm.cdb.serviceinfrastructure.compatAPI.preserveFunctionalGroups**

Determines the process of naming functional groups when business applications that are compatible with earlier versions are created.

If you set this property to *true*, functional groups have the same names as tiers names. However, the following conditions must be also met:

- The migrated grouping pattern must have the *useMigratedTierNames* attribute set to *true*.
- The selector must have the *tierName* attribute set.

If you set this property to *false*, functional group names are set by using the grouping pattern configuration.

The default value is *true*.

For more information, see [“Business application tiers” on page 235](#), and [“Tiers configuration” on page 207](#).

**com.ibm.cdb.internalscheduling.bizapps.<GROUP NAME>.enabled**

Determines whether to enable or disable the provided execution group on the server.

The allowed values are *true* and *false*.

**com.ibm.cdb.internalscheduling.bizapps.<GROUP NAME>.threads**

Specifies the number of available threads for the provided execution group.

The value is specified in numbers, for example 5.

**com.ibm.cdb.internalscheduling.bizapps.schedule name.misfire**

Defines how long a pattern can wait to be run before it is skipped from processing and a warning is logged. The warning contains the information that the pattern did not start. To prevent it, you can create more thread pools, or modify the schedules of your patterns.

The default value is 60, expressed in minutes.

This property is defined per schedule. For example, if you want to change the value of this property for the schedule named *default*, modify the name of the property in the following way:

```
com.ibm.cdb.internalscheduling.bizapps.default.misfire
```

If you do not want to modify your configuration and at the same time you do not want the patterns to be skipped, set this property to a higher value.

**Note:** You can also adjust this setting by using the *misfireLimit* option of the *createSchedule* action of the *bizappscli* tool. For details, see [“Actions for managing grouping pattern schedules” on page 222](#).

## Logging

When business applications are generated, log files are also created and stored in the `$COLLATION_HOME/log` directory. When you have problems that are related to business applications, log files can help you to troubleshoot them.

There are two kinds of logs created for business application.

### Logs created when working with grouping patterns and business application in Data Management Portal

- Log messages are in the `log/tomcat.log` file (TADDM 7.3.0) and in the `log/wlp.log` file (TADDM 7.3.0.1, and later).
- Error messages are in the `log/error.log`, `log/tomcat.log` (TADDM 7.3.0) and `log/wlp.log` (TADDM 7.3.0.1, and later) files.

### Logs created when working with business application composition engine and running patterns from the pattern list

It is advised to enable split logging for business applications composition engine, so that the logs are split into separate files. It improves readability. To enable split logging, set the `com.ibm.taddm.log.split.bizapp` property to `true` in the `collation.properties` file. If this property is set to `false`, all log files are placed in the `log/services/PatternsSchedulingService.log` file.

- Log messages are in the following directories:
  - `log/services/PatternsSchedulingService.log` file - contains general logs from scheduling engine, without information from particular patterns.
  - `dist/log/bizapps/<pattern>/<starttime>.log` - contains separate log files for every grouping pattern and their processing time. The `<pattern>` folders are created for each grouping pattern in the format `[pattern name-GUID]`, for example `J2EE App pattern-F12AC23451AB3A4FAF58E9187ABF1169`. Inside the pattern folders, you can find log files created for every grouping pattern processing in the format `[time of generation in milliseconds].log`, for example `1416408057548.log`.
- Error messages are in the `log/error.log` file and in the same files that contain log messages.

To change the log level, modify the `com.collation.log.level.vm.Topology` property in the `collation.properties` file. To change the INFO level, you can also use the grouping patterns configuration. For more information, see the *Logging properties* topic in the *TADDM Administrator's Guide* and [“General configuration”](#) on page 202.

## Migration from 7.2.2 and automatic conversion of old business applications

When you upgrade from TADDM 7.2.2 to a higher version, all business applications are converted automatically into grouping patterns in accordance with a specific set of rules.

### Automatic conversion

A new grouping pattern is created for each application. Later, the `BizAppsAgent` generates new custom collections from these grouping patterns. The new custom collections' content (a set of configuration items) is the same as the content of the corresponding original collections.

**Note:** Business applications can contain only high-level and middle-level objects. Therefore, some CI types, which were high-level objects in TADDM 7.2.2, are no longer high-level objects in version 7.3.0. As a result, they are not added to business applications. The new high-level types are `SComputerSystem`, `SSoftwareServer`, `SLogicalGroup`, `SPhysicalFile`, `SSoftwareInstallation`, `SFunction`. The new middle-level type is `SDeployableComponent`. Additionally, there is no `OperatingSystem` type in the new model. Its attributes were merged into the `simple.SComputerSystem` class.

The following rules apply when old business applications are converted into grouping patterns:

- Pattern type is set according to a group type.

- The groups of the business application, and business service types are converted into grouping patterns of the business application type.
- The groups of the collection type are converted into grouping patterns of the collection type.
- The groups of the access collection type are converted into grouping patterns of the access collection type.
- Selectors of a new grouping pattern are created to reflect the content of the corresponding old group.
  - The MQL query selector is created for each rule in a group.
  - The instance-based selector is created and it contains all CIs that are manually selected as the content of the group.
  - The selectors do not use Dependency Traversal Template, so that only CIs selected by each selector are included, without automatically including dependencies.
  - Functional groups are converted into separate selectors with a tier name set to the functional group name. Tiers are not visible for users, but they are used during creating business entities compatible with earlier versions to recover original functional groups structure.
- Grouping name expression is set to the name of the original collection so that the new custom collection has the same name.
- Old migrated grouping entities are not removed from the database.

**Note:** If an error occurs during the migration, the grouping entities are migrated anyway. The affected grouping pattern is marked with an exclamation point on the list, and the error description is displayed in the **Edit** window. All invalid selectors are also marked.

## Compatibility of new custom collections with earlier versions

Some model types present in earlier versions of TADDM are deprecated in the new approach to business applications. These types are Application, Collection, AccessCollection, BusinessSystem, and FunctionalGroup. However, you can create objects of these types to ensure compatibility.

To enable the compatible conversion, use the `com.ibm.cdb.serviceinfrastructure.earlier.ver.compatibility` property in the `collation.properties` file. The default value of this property is *TRUE* for the upgrade scenario and *FALSE* for the fresh installation scenario.

**Note:** The groups of the old type are not displayed in the UI. They can be listed only by using API.

To ensure that the content before and after the conversion is the same, special tiers (TADDM 7.3.0) or manual tiers (TADDM 7.3.0.1, and later) that are linked to selectors are introduced. The tier names are the same as the functional group names. Functional group names that are defined in application descriptor files are also converted into special or manual tiers. Thanks to the compatibility conversion, the following elements are preserved:

- The type of old grouping entities. It is vital to differentiate between business services and business applications because both are converted into a pattern of the business application type.

**Note:** The following limitation is valid in TADDM 7.3.0 only, it is not applicable for TADDM 7.3.0.1, and later.

This function refers only to migrated entities, so only migrated grouping patterns can be converted back into business services. Manually created grouping patterns of the business application type are always converted back into old business applications.

- The structure of business application functional groups.

**Note:** The following limitation is valid in TADDM 7.3.0 only, it is not applicable for TADDM 7.3.0.1, and later.

However, one CI can be assigned to only one tier, so in case of old business applications compatible with earlier versions, a single CI cannot be shared between several groups. If one CI is shared between several groups, it is removed from all groups except one.

## Business application tiers

Business application tiers are groups of similar business application elements. They are used to create functional groups to integrate TADDM with other products and to ensure compatibility with business applications that were created in TADDM 7.2.2.

Functional groups that are created from tiers have the same names as the tier names. Tiers are created based on the business traits of business application elements. For example, one tier can contain all computer systems. Another tier can contain all application servers. And the third tier can contain all Java Platform, Enterprise Edition applications. Assigning CIs to particular tiers is based on rules that are defined by users in configuration files.

### TADDM 7.3.0

#### Special tiers

There are two kinds of tiers, regular and special. Special tiers are designed exclusively for the following kinds of business applications:

- The ones which were migrated from TADDM 7.2.2.
- The ones which were created from application descriptors and which have functional group names set.

Tiers are used to create functional groups with the same names as tier names only if all of the following conditions are met:

- The `com.ibm.cdb.serviceinfrastructure.compatAPI.preserveFunctionalGroups` property is set to `true` in the `collation.properties` file.
- The migrated grouping pattern has the `useMigratedTierNames` attribute set to `true`.
- The selector has the `tierName` attribute set.

If the preceding conditions are not met, tier names and, as a result, functional group names are set by using the grouping pattern configuration. For more information, see [“Tiers configuration” on page 207](#).

Special tiers have the following characteristics:

- Special tiers have precedence over regular tiers. If a special tier exists, the regular tier name is ignored.
- Special tiers apply only to the core CI (migrated business applications by default contain only core CI, they have no Dependency Traversal Templates).
- Special tiers can be turned off globally by setting the `com.ibm.cdb.serviceinfrastructure.compatAPI.preserveFunctionalGroups` parameter to `false` in the `collation.properties` file.
- Special tiers can be turned off per single grouping pattern in Data Management Portal.

#### **Fix Pack 1** TADDM 7.3.0.1

There are two kinds of tiers, regular and manual.

#### Regular tiers

Regular tier names are calculated by using the grouping pattern configuration. For more information, see [“Tiers configuration” on page 207](#).

#### Manual tiers

You can set manual tiers for selectors manually in Data Management Portal, in Grouping Patterns pane. However, manual tiers are set automatically for the following kinds of grouping patterns:

- The ones, which were migrated from TADDM 7.2.2.
- The ones, which were created from application descriptors and have functional group names set.

Manual tiers have precedence over regular tiers. If a manual tier exists, the regular tier name is ignored.

## Special cases in the automatic conversion of old business applications

During the automatic migration and conversion of the old grouping entities like business applications, collections, access collections, and business services, special cases might occur.

### Wrong MQL rules

During the migration of MQL rules, all queries are tested for validity. If an MQL query is invalid, additional instance-based selector is created in the grouping pattern. The selector contains all CIs from the migrated group. MQL-based selector with invalid MQL query is also created, but is marked as disabled and is not processed by the agent. Grouping patterns with invalid selectors are highlighted in Data Management Portal, allowing users to easily find and correct invalid MQL queries.

**Note:** Wrong MQL queries are rare. However, due to Model Changes in the latest TADDM version, some MQL queries might be no longer valid.

### MQL rules and the content of business applications

Migration algorithm executes the MQL rule of every business application (or functional group) and compares the results with the business application content. All CIs that are a part of the business application content but were not returned by any of the MQL queries are included into the instance-based selector.

**Note:** This algorithm is used only if the **Replace all existing content with the new content** flag is not set.

### MQL rules for old business applications

MQL rules that operate on old business applications, for example "select \* from Collection", are treated like rules with invalid MQL queries. The following two selectors are created for such rules:

- MQL selector that contains the same MQL query. It is marked as invalid and is disabled.
- Instance-based selector that contains all resulted CIs.

It is caused by the difference between the new model based on grouping patterns and old business applications model. Therefore, it is impossible to automatically convert an MQL query that returns old business applications into a query that returns corresponding custom collections.

### Old business application deprecation

All types of grouping entities are automatically migrated into grouping patterns during the upgrade from TADDM 7.2.2. After the migration, they are not deleted because of requirements to ensure compatibility with an earlier version (they are needed for integration with external systems). They are deprecated and hidden from a user. A user can perform only a narrowed set of operations on them through API.

### Automated linking mechanism

Automated linking mechanism ensures consistency of relations between the migrated custom collections and the original business applications.

Business applications can be parts of other business applications. For example:

- A business service can contain applications.
- An access collection can contain collections.
- An application can contain business services.

After conversion of business applications to new custom collections through grouping patterns, new custom collections are contained in all the places where the old business applications were contained.

#### Example

1. Business application with the name BA1 (BusinessApplication:A1) contains collection C1 (Collection:C1).



2. During the migration phase of the installation, new grouping patterns (GroupingPattern:BA1 and GroupingPattern:C1) are automatically created basing on old business applications.
3. The newly generated CustomCollection:BA1 contains CustomCollection:C1.

## Empty business applications migration

The core function of business applications is to group the existing infrastructure elements into one entity that has common business designation. The new model based on grouping patterns does not allow defining empty custom collections because they are pointless regarding their designation. However, the old business application model allowed creating empty business applications. In order not to lose any data, TADDM is also able to convert empty business applications.

To convert empty business applications, a grouping pattern is created. It is then configured to create a custom collection that contains only the grouping pattern itself. However, grouping patterns cannot be included in the custom collection content. Therefore, the grouping pattern is automatically removed and the custom collection is empty.

This is only a mechanism ensuring compatibility with earlier versions and is not intended to be used deliberately. Empty business applications are often created because of misconception about business applications. For example, business applications are not intended to represent applications that are conceived as Operating System applications (programs).

## Instance-based selectors in a grouping pattern

Business applications can have the following content:

- Content that is not covered by MQL rules or covered by invalid MQL rules.
- Content that is covered by MQL rules but of business application type.

Such content is placed into instance-based selectors.

For business services, collections, and access collections, one instance-based selector is created per group. For business applications, instance-based selector is created per every functional group.

## Integrating business applications with other Tivoli products

You can integrate business applications with other Tivoli products.

A new function was introduced to enable integration between TADDM and products that read data from TADDM by using DataApi or directly from TADDM database by using SQL. To learn about this function, see the *Business entities compatibility with earlier versions* topic in the *TADDM Administrator's Guide*.

## Example scenarios

Learn how to create one business application from several selectors, and many business applications from one grouping pattern, and how to customize grouping pattern configuration.

### Creating simple business application and adding manual dependencies

Learn how to create a simple business application based on Java Platform, Enterprise Edition application, and how to add manual dependencies between the components.

In the following example, a database with sample data is used. The database contains the following components:

- WebSphere Cell with some servers and deployed Java Platform, Enterprise Edition applications
- JBoss servers with some applications
- Computer systems, of which some are a part of virtual environment
- DB2 databases
- Web servers

- other components

The first aim of this task is to create a simple business application based on a Java Platform, Enterprise Edition application. The second aim of this task is to add components to the existing business application, either by creating another selector, or adding manual dependencies between components.

1. Open Data Management Portal.
2. In the Functions pane, click **Discovery** if you use TADDM 7.3.0, or 7.3.0.1, or **Analytics**, if you use TADDM 7.3.0.2, or later.
3. Click Grouping Patterns. The **Grouping Patterns** table is displayed.
4. In the name field, enter EDay Trader.
5. Set the pattern type to Business Application.
6. For the **Schedule** option, choose Every Saturday night.
7. In the **Description** field, enter EDay Trader application and click **Next**.
8. Click **New** to add a selector.
9. In the **Name** field, enter EDay Trader's J2EE App.
10. In the **Description** field, enter EDay Trader J2EE application, and click **Choose**.
11. In the query editor, select Instance-based as selection type.
12. Browse your database, select the components that you want to be your core CIs, click **Add**, and then click **OK**.
13. Optional: To see the results, click **Test**.
14. Click **OK**. You are in the **Create a new Grouping Pattern** window again.
15. To enable data traversing, make sure that the **Use Dependency Traversal Template** check box is selected. Select the **HigherDown** and **Lowerup** options, and clear the remaining ones.
16. Click **Finish**.

Generating business application manually

17. To see the new business application, you do not have to wait for the scheduled time of generation. You can generate the application manually.
  - a) Open the Grouping Patterns pane.
  - b) Select the grouping pattern that you created and click **Execute**.
  - c) Wait for a while and then click **Refresh View**.
  - d) To view the new business application, go to the Discovered Components pane.

Creating a new selector and adding it to the EDay Trader business application.

18. If you notice that some components are not part of the business application that you created, you can add them by creating a new selector. The missing elements for EDay Trader business application are DB2 database and Web Server. First, create a new selector with DB2 database used by EDay Trader application.
  - a) In the Grouping Patterns pane, select the EDay Trader pattern and click **Edit**.
  - b) Go to the **Selectors** tab and click **New**.
  - c) In the **Name** field, enter EDay Trader database and click **Choose**.
  - d) Choose the core CI the same way as in step 11 and 12. Add the missing database.
  - e) Do not change grouping name expression, it must generate the same name as the previous selector. In this way, compositions created from these two selectors are merged into one business application. Click **OK**.
  - f) To confirm the changes, click **OK**.
  - g) To see the updated business application, regenerate the grouping pattern and go to the Discovered Components pane.

Creating manual dependencies

19. Web server is still not part of the new business application, and there is no connection between Java Platform, Enterprise Edition application and the database. However, instead of creating many

selectors, you can create manual dependencies between Java Platform, Enterprise Edition Application and DB2 database, and between Java Platform, Enterprise Edition application and Web Server.

- a) Add Java Platform, Enterprise Edition application to the cart by right-clicking it in the topology and selecting **Add to cart** option.
- b) In the Discovered Components pane, click **Cart** and select the application check box.
- c) From the **Actions** list, select **Show dependencies**.
- d) Click **New**, select **Dependent** and browse for the database that you want to add. Click **OK**.
- e) To add the dependency for Web Server, click **New**, select **Provider** and browse for the server. Click **OK**.
- f) Click **Close**.

**Note:** With the manual dependencies created, you no longer need the additional selector that you created in step 18. Go to the Grouping Patterns pane, select EDay Trader pattern and click **Edit**. In the **Selectors** tab, select the EDay Trader database selector and click either **Delete** to remove it, or select the **Disabled** check box, so that it is not processed.

## Creating many business applications from one grouping pattern

Learn how to create a universal grouping pattern that generates many business applications automatically.

In the following example, a database with sample data is used. The database contains the following components:

- WebSphere Cell with some servers and deployed Java Platform, Enterprise Edition applications
- JBoss servers with some applications
- Computer systems, of which some are a part of virtual environment
- DB2 databases
- Web servers
- other components

The aim of this task is to create a separate business application for each Java Platform, Enterprise Edition application that is deployed on every application server in the example environment. You do not have to create separate grouping patterns for them, only one is enough. Moreover, when you add an application to the database after you created the grouping pattern, a business application is still automatically created for such application.

1. Open Data Management Portal.
2. In the Functions pane, click **Discovery** if you use TADDM 7.3.0, or 7.3.0.1, or **Analytics**, if you use TADDM 7.3.0.2, or later.
3. Click **Grouping Patterns**. The **Grouping Patterns** table is displayed.
4. In the name field, enter Generic J2EE applications.
5. Set the pattern type to Business Application.
6. For the **Schedule** option, choose Every day at 1am.
7. In the **Description** field, enter Generic J2EE applications and click **Next**.
8. Create new selector. In the **Name** field, enter Generic J2EE Selector.
9. In the **Description** field, enter Selector creating separate Business Application for every J2EE application, and click **Choose**.
10. In the query editor, select the MQL query as the selection type and in the **Query** field enter `J2EEApplication WHERE domain is-null`.
11. The grouping name expression resolves to a business application name, which is a unique key that identifies business applications. Therefore, it cannot resolve to a pattern name, as only one business application would be created. To avoid it, you can use the Java Platform, Enterprise Edition

application name as the grouping name expression. In the **Grouping Name Expression** field enter the following expression:

```
${coreCI.displayName}
```

To see the results, click **Test**.

The names are long. To shorten them, instead of the `displayName` attribute, you can use the `name` attribute:

```
${coreCI.name}
```

To see the results, click **Test**.

You can further change some of the names, for example, the ones which have ear file name. You can remove the `.ear` extension by using the following expression:

```
$utils.regex(${coreCI.name}, "(.*)\\.ear", "(.*)" )
```

In this expression, the `"(.*)\\.ear"` expression removes the `.ear` extension from the file name, and the resulting name is the remaining part of that file name. The `"(.*)"` expression leaves the name as it is. To see the results, click **Test**.

You do not need to modify the names any further. However, there are applications with the same name from different environments, for example the test and production environments. To distinguish them, you can add to their names a prefix with JBoss domain name, or WebSphere Cell name. Use the following expression:

```
$utils.or($coreCI.parent.parent.name, $coreCI.parent.parent.displayName)
$utils.regex(${coreCI.name}, "(.*)\\.ear", "(.*)" )
```

WebSphere Cells have always the `name` attribute set, so the `$coreCI.parent.parent.name` expression returns all WebSphere Cell names. It is not always the case for JBoss domain, where sometimes the `displayName` attribute is set. Use the `or` expression so that the first argument that is not null is returned for each application. To see the results, click **Test**.

You can still have more information, for example server type and a prefix to easily distinguish Java Platform, Enterprise Edition based applications from others. To add the prefix, enter J2EE in front of the expression:

```
J2EE $utils.or($coreCI.parent.parent.name, $coreCI.parent.parent.displayName)
$utils.regex(${coreCI.name}, "(.*)\\.ear", "(.*)" )
```

To add the server type, use the following expression:

```
J2EE [${coreCI.parent.productName}] $utils.or($coreCI.parent.parent.name,
$coreCI.parent.displayName)$utils.regex(${coreCI.name},
"(.*)\\.ear", "(.*)" )
```

The server type is taken from the `productName` attribute of application servers. Because application servers are parents of Java Platform, Enterprise Edition applications, the expression that generates server type is in square brackets. To see the results, click **Test**.

The server type names are too long. You can remove the `Application` word from the server type name by using the following expression:

```
J2EE [$utils.regex($coreCI.parent.productName, "(.*) Application.*", "(.*)")]
$utils.or($coreCI.parent.parent.name, $coreCI.parent.displayName)
$utils.regex(${coreCI.name}, "(.*)\\.ear", "(.*)" )
```

If the name contains the `Application` word, it is removed by the `"(.*) Application.*"` expression. If the name does not contain such word, the `"(.*)"` expression leaves it as it is. To see the results, click **Test**.

The results look good. Click **OK**.

12. To save the new grouping pattern, click **Next** and then **Finish**.

To see the new business applications, open the **Grouping Patterns** pane, select the grouping pattern that you created and click **Execute**. Wait for a while and then click **Refresh View**. After the grouping pattern is generated, you can view the new business applications in the **Discovered Components** pane.

## Configuring a grouping pattern configuration

Learn how to customize a business application by configuring tiers, reversing relations, and importing configurations by using the `bizappscli` tool.

The environment in this example scenario contains the following elements:

- IIS web services version 7
- IIS modules that are deployed on the IIS web services
- Oracle instances
- Computer systems on which the application servers run.

The aim of this task is to create a business application that shows relations of IIS modules, and to map these relations to the IIS web services on which these modules are deployed. The relations that are covered by this application are:

- Transactional dependencies of IIS modules
- Relations of the parent IIS web services to these modules
- Relations of all application servers that are found by the grouping pattern of this application to the computer systems on which these application servers run.

**Note:** Fix Pack 2 The relation between IIS module and Oracle instance is supported for TADDM 7.3.0.2, and later.

At the same time, IIS modules are excluded from the business application, but their relations are mapped to IIS web services.

Additionally, the elements of this business application are grouped into three tiers - the first for IIS web services (BL tier), the second for Oracle instances (DB tier), and the third for all other elements, including computer systems (common functional group tier). The tier names are specified to create functional groups to enable integration with products like Tivoli Business Service Manager, where functional groups are required.

1. Reverse the direction of traversal so that the relations between IIS web services and Oracle instances are discovered. IIS modules have outgoing relations to both IIS web services and Oracle instances. If the IIS modules are filtered out, the relation between the IIS web service and Oracle instance cannot be created. However, you can reverse the direction of traversal so that IIS web service has outgoing relation to IIS module. In such case, even if the module is excluded from the application, the traversal is not stopped, and the relation between IIS web service and Oracle instance is created.

To achieve this, you must modify the direction configuration. Since you cannot modify the direction configuration for a chosen grouping pattern, you must modify the default configuration that applies to all grouping patterns.

- a) Export the default configuration by using the `bizappscli` tool. You do not have to export the whole configuration. You can export only the direction configuration.  
From the `<taddm installation directory>/dist/sdk/bin` directory, run the `bizappscli` tool with the following options:

```
bizappscli.sh exportDefaultConfiguration -d -f default_directions.xml
```

where the `-d` option specifies to export only the directions configuration, and the `-f default_directions.xml` option creates a file, which you can edit. The XML file is created in the `dist/sdk/bin` directory.

- b) Open the `default_directions.xml` file. Modify the end of the file so that the `DeployedTo` relations are the same as in the following example:

```
<reverse relation="only relation.DeployedTo" source="simple.SLogicalGroup"
```

```
target="simple.SDeployableComponent"/>
<reverse relation="only relation.DeployedTo" source="app.SoftwareModule"
target="app.AppServer"/>
```

Save the changes.

2. Create a custom configuration for your grouping pattern. Create an XML file and name it, for example, `config.xml`. Add the following content to your file:

```
<xml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="defaultPatternConfiguration.xsd">
  <general>
    <maxHopsLimit>2</maxHopsLimit>
    <firstTierOnly>true</firstTierOnly>
    <infoLevel>GENERAL</infoLevel>
  </general>
  <compositionConfiguration>
    <excludeFromComposition>
      <exclude type="com.collation.platform.model.topology.app.web.
iis.IIsModule"/>
    </excludeFromComposition>
  </compositionConfiguration>
  <tiers>
    <tier>
      <name>DB</name>
      <rule>
        <className>OracleInstance</className>
      </rule>
    </tier>
  </tiers>
  <traversalConfiguration>
    <excludedRelationships>
      <exclude relation="{any}" source="{any}" target="{any}"/>
    </excludedRelationships>
    <includedRelationships>
      <include relation="only relation.DeployedTo" source="app.web.
iis.IIsModule" target="app.web.iis.IIsWebService"/>
      <include relation="app.dependencies.Transactiona
Dependency" source="app.web.iis.IIsModule" target="app.db.oracle.
OracleInstance"/>
      <include relation="only relation.RunsOn" source="app.AppServer"
target="sys.ComputerSystem"/>
    </includedRelationships>
  </traversalConfiguration>
</xml>
```

**Note:** Fix Pack 1 The `firstTierOnly` option is available in TADDM 7.3.0.1, and later.

These configuration sections have the following functions:

- `<general>` - the `maxHopsLimit` parameter specifies that 2 levels of relations are traversed, starting with IIS modules because data traversal template is enabled only for the selector that queries IIS modules.
- `<compositionConfiguration>` - excludes the IIS module from the business application.
- `<tiers>` - specifies that all objects that are found on the Oracle instance are a part of the tier named DB.
- `<traversalConfiguration>` - excludes all relations from being traversed, except the relations between IIS modules and IIS web services, between IIS modules and Oracle instance, and between all servers and their hosts.

3. Import all your changes by running the `bizappscli` tool in the following order:

- a. Import the default direction configuration:

```
bizappscli.sh importDefaultConfiguration -f default_direction.xml
```

- b. Import the custom configuration:

```
bizappscli.sh importConfiguration -U -c custom_config -f config.xml
```

where:

- `-U` - updates the existing configuration if you are importing a modified file instead of a new one.

- `-c custom_config` - defines the name of your custom configuration.
4. Create a grouping pattern and define two selectors.
    - a) Open Data Management Portal.
    - b) In the Functions pane, click **Discovery** if you use TADDM 7.3.0, or 7.3.0.1, or **Analytics**, if you use TADDM 7.3.0.2, or later.
    - c) Click **Grouping Patterns** and then **New**.
    - d) Specify the name, pattern type, and schedule.
    - e) From the **Configuration** list, select **custom\_config**. It is the customized configuration that you created for your grouping pattern. Click **Next**.

**Note:** Alternatively, when you created the grouping pattern, you can attach the customized configuration to it by running the `bizappscli` tool with the following options from the `dist/sdk/bin` directory:

```
bizappscli.sh attachConfiguration -c custom_config -n my_pattern | -g
00000000000000000000000000000000
```

where `my_pattern` is the name of your grouping pattern and `00000000000000000000000000000000` is the GUID of your grouping pattern. Use either the `-n` or the `-g` option.

- f) Create the first selector. In the **Selector name** field, type `BL_module`.
- g) **Fix Pack 1**  
Leave the **Tier name** field empty. If you specify any value in this field, all objects that are found by this selector are assigned to this tier. The IIS modules are excluded from the business application, and therefore they do not have to be a part of any tier. However, if this field is empty, but you attached a custom tier configuration to your grouping pattern, the configuration applies. All objects that are found by the selector and that meet the requirements that are specified in the tier configuration are assigned to the specified tier. In this case, as specified in the `custom_config` configuration, all objects that are found on the Oracle instance are assigned to a tier named `DB`.
- h) Specify the following MQL query:

```
SELECT * FROM IISModule WHERE ((lower(name) == '/bl')) and (!(parent.
productVersion) starts-with '7')
```

This query looks for IIS modules that have the `'/bl'` name and are a part of IIS web service version 7.

- i) In the **Grouping name expression** field, type `My App`.
- j) Enable the **Data Traversal Template** option and select all traversal options to find dependencies between IIS modules and other objects.
- k) Create the second selector. In the **Selector name** field, type `BL`.

- l) **Fix Pack 1**  
In the **Tier name** field, type `BL`. All IIS web services that are found by this selector are a part of this tier.

- m) Specify the following MQL query:

```
SELECT * FROM IISWebService WHERE (exists (lower(modules.name) == '/bl'))
and (!(productVersion) starts-with '7')
```

This query looks for the parent IIS web services of the modules that are specified for the first selector.

- n) In the **Grouping name expression** field, type `My App`.
- o) Disable the **Data Traversal Template** option. As a result, no IIS web services dependencies are found. However, the hosts of these IIS web services are found by the query that is specified for the first selector, because all traversal options are selected.

- p) Click **Next**.
- q) Optional: Specify Administrative Information.
- r) Click **Finish**.

Wait for the scheduled run of your grouping pattern. Alternatively, you can run it manually by selecting it from the grouping patterns list in Data Management Portal and clicking **Execute**. After a while, you can display the topology in Data Management Portal in **Topology**.

#### **Related reference**

[“Grouping patterns configuration” on page 200](#)

You can control the process of business application creation by using grouping patterns configuration. The configuration allows you to include or exclude chosen relations during data traversal and chosen classes from resulting business applications. You can also change the dependency direction for relations that are defined in Common Data Model, and assign tiers to business application elements.

[“bizappscli tool” on page 216](#)

You can use the CLI `bizappscli` tool to manage grouping patterns, grouping pattern processing schedules, grouping pattern configurations and the execution of grouping patterns.

**Fix Pack 2** You can use the tool to create reports for analyzing the content of business applications.

**Fix Pack 3** You can use the tool to export the graph of the business application topology to the SVG format.



## Notices

---

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
224A/101  
11400 Burnet Road  
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

## Trademarks

---

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe is either a registered trademark or a trademark of Adobe Systems Incorporated in the United States, and/or other countries.



Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.



